A METHODOLOGY FOR THE DESIGN OF AN ASSEMBLY LINE WITH VARIABLE TASK TIMES

By
SUBHASH CHANDRA RASTOGI

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
OCTOBER, 1973

A METHODOLOGY FOR THE DESIGN OF AN ASSEMBLY LINE WITH VARIABLE TASK TIMES

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
SUBHASH CHANDRA RASTOGI

to the
DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
OCTOBER, 1973

TO MY PARENTS

5.11.73
24

CERTIFICATE

This is to certify that the work 'A Methodology for the Design of an Assembly Line with Variable Task Times' by Subhash Chandra Rastogi has been carried out under my supervision and that this has not been submitted elsewhere for a degree.

DR. J.L. BATRA
Assistant Professor
Department of Mechanical Engineering
Indian Institute of Technology Kanpu

ACKNOWLEDGEMENT

I take this opportunity to place on record my indebtness to Dr. J.L. Batra for his invaluable help and guidance.

Help and suggestions coming from Dr. S. Ramasheshan and Mr. S.D. Kothari are gratefully acknowledged.

I am thankful to the management and staff of J.K. Electronics, Kanpur for providing the necessary data.

Thanks are due to so many of my friends who have helped me in more ways than one.

Last but not the least, I am thankful to Mr. Uma Raman Pandey for typing the manuscript neatly and patiently.

SUBHASH CHANDRA RASTOGI

TABLE OF CONTENTS

SYNOPSIS

SUBHASH CHANDRA RASTOGI
M.Tech.(Mech.)
Indian Institute of Technology Kanpur
October, 1973

A METHODOLOGY FOR THE DESIGN OF AN ASSEMBLY LINE WITH VARIABLE TASK TIMES

In the present study a design methodology which accounts for the following important features of assembly line production has been developed :

1. A case where the time requirement for at least one task is more than the cycle time,
2. Variability in the task times,
3. Failure characteristics of the work stations.

The design methodology has been divided into two phases. The first phase considers only the first feature whereas the other two features are taken care of in the second phase.

The first phase consists of allocation of the total number of the tasks of the line among the various work stations, in order to achieve maximum compactness and uniformity in the distribution of total work load. The task times have been considered deterministic in this phase. The case, where the cycle time is less than the task time for at least one task of the assembly line, has been incorporated in this phase by introducing parallel stations. COMSOAL technique has been

used to achieve the maximum compactness and the concept of smoothness Index is used to achieve maximum uniformity.

The second phase accounts for the variability in the task times as well as the failure of the work stations. Buffer inventories are introduced to account for both of these features. In this phase the optimum levels of the buffer capacities are determined which minimize the total cost of production per unit produced from the line. Digital Simulation and Univariate search techniques have been used to achieve the optimum size of the buffer capacities.

The proposed methodology has been computerized and validated with the help of the data collected from an electronics industry manufacturing television sets.

CHAPTER I

INTRODUCTION

With the advancement of civilization and consequent enhancement in the standard of living of the people in general, there have been great changes in the demand patterns and functional requirements of all sorts of commodities. The demand curves for both civilian goods and military equipments have been rising in all its multitudes. This posed a big challenge to the managers and technicians all over the world to revise the production system radically as to make them capable of meeting the increased demands of varied nature of commodities. Research scholars put their heads together to devise ways and means for achieving mass scale production. These efforts resulted in technologically improved system suited for mass production. Line system of production is one such advancement whose value was recognized during the period of transition from small scale to large scale production.

Henry Ford is credited for being the first person to implement the line system of production in its highly developed form for Ford Highland Park Plant. This was in 1913 [8]. It was observed that this system brings out a sharp increase in productivity and also improves the quality of the product at reduced cost of production.

The method was found so attractive and useful that it has been adopted by all kinds of industries ranging from toys to automobiles.

The line system of production, as Demyanyuk [8] puts it, is a production system where the machining or assembling operations are allotted to definite pieces of equipments and operators. The operations required by the product are arranged according to their precedence only. The semi-finished products are moved from one work station to the next after completion of the preceding operation.

Among the problems associated with line system of production, the line balancing is the most important one. The essence of line balancing problem lies in apportioning the total work load required for assembly work among the work stations uniformly and compactly following the precedence restrictions.

This problem generally involves either,

(1) minimization of number of stations for a specified output rate,

or (2) maximization of the output rate for a fixed number of work stations.

A number of analytical and heuristic models have been suggested for the purpose of balancing a line for a required output rate. A few research workers have concerned

themselves to the second criterion. The models available in literature are significantly different from the real life situation of an assembly line. Moreover, the objective function in most of the available models is the minimization of total idle time of the assembly line. This does not guarantee the minimization of total cost of production, which is of greater interest to the industries. The factors which have been either overlooked or have not been paid due attention are listed as follows :

1. The random variability of the station times
2. Uncertainties related with the failure of the stations
3. A situation where at least one task time is so large in its time content that it cannot be performed by a single operator to fulfil the required output rate.

These factors very much govern the actual throughput and cost of operation of the line, and should be taken into account to represent an actual line realistically. The present work has been directed to develop a model which considers each of the above listed factors. Interstage inventories have been introduced in between each pair of successive stations to absorb the effect of variability of the task times. The problem has been divided into two phases :

<u>Phase I</u> : Maximization of output rate by balancing the line, ignoring the effect of variability of task time.

<u>Phase II</u> : Determination of optimal levels of interstage inventories in between every pair of successive stations, taking into account the variability of task time and failure of stations.

The model, incorporating above listed two phases, has been programmed in FORTRAN IV language for use on IBM 7044 digital computers. The model has been applied to an assembly line of J.K. Electronics, Kanpur. Although the model has been primarily tested for the said television industry, every care has been taken to make the model general in nature and, therefore, may be adapted for other assembly lines without much difficulty.

CHAPTER II

LITERATURE REVIEW

The need for balancing the production line was felt as early as in 1900 but a successful technique to achieve this was not found until 1913. Since then much efforts have been put in this direction and as many as twenty models have come up during the last couple of decades. Some models take a realistic approach to solve practical problems, while others deal with its theoretical structure with a view to gain deeper understanding of the problem without making any attempt towards their application to large scale balancing problems.

The first paper published in this field was by Salveson [33] in 1955, though credit should have gone to Bryton [4], who was first to analyse this problem analytically in 1954. Bryton's research work was not published however. Salveson's paper suggests that one should enumerate all possible work stations and then use Linear Programming to choose the combination of stations that best meets the requirement placed on the solution. The problem was again formulated as a Linear Programming Problem (this time as an Integer Programming Problem) by Bowman [5], in 1960. Salveson's method is unacceptable for it results in fractional value of work stations which cannot be achieved in reality. This difficulty is overcome in Bowman's model.

Bryton's methodology suggests interchanging the pair of tasks between the largest and the smallest stations (on time scale) according to a specified rule, to attain a better balance. The simple concept of each transfer causing improvement seems rather unconvincing, and he himself admits it. Jackson's enumerative procedure [18], published in 1956, promises an optimal solution, but it requires enormous computational efforts while applying to real life problems. In fact, all the models mentioned above need too much of computational efforts to be practical and therefore, are of little value to industries.

In 1961, Kilbridge and Wester [19] proposed a technique which is found useful for a variety of line balancing problems. It is a heuristic technique and requires little ingenuity and little knowledge of arithmatic. The method makes a systematic use of the precedence graph and appears to be a useful way for arriving at a balanced line by hand computations. Comparison of its ability with other techniques has been done later in this chapter.

In the same year, another procedure, popularly known as RPW (Ranked Positional Weight Technique), was proposed by Helgeson and Birnie [14]. The method is logically very simple and needs only one iteration. It is workable for small lines but significantly deviates from the optimal solution for large size problems. An improvement to this method was suggested by Mansoor [24] in

1964. He claims that an optimal solution is guaranteed if proposed modifications are followed, but the amount of work required may be quite large.

In 1963, Held, Karp and Sherishian [13] developed another analytical procedure using Dynamic Programming approach. This has a limitation on the size of the problem it can handle, otherwise it is the best technique available so far, for line balancing problems. To cope up with the large size problems, the authors have suggested a method to group some tasks and consider them as a single task. This modification reduces the size of the problem, however, it does not guarantee an optimal solution. The capabilities of this technique are discussed in details later in this chapter.

In 1963, another model, called Successive Maximum Elemental Time Technique, was introduced by Hoffman [15]. This model develops a somewhat modified form of precedence matrix which is then used to generate all feasible permutations of tasks and assigns them to the various stations. It gives only an approximate solution, as was evaluated by Ignall [17] and others [21, 27].

In 1965, Tonge [35] came out with some probabilistic combinations of heuristics for choosing the next task to be assigned to the station. Two other important methodologies, for balancing the line, were developed by Moodie and Young [28], in 1965, and by Arcus [1], in 1966. It is

hoped that the former technique provides good assembly line balances for both manual and computer mode of solution, but not much pain has been taken by the research workers to verify this, while other methods have been tested for their effectiveness by more than one author. Arcus's method, popularly known as COMSOAL (Computer Method of Sequencing Operations for Assembly Line), is a heuristic method and is claimed to be best for large size line balancing problems [26].

Some other methods for line balancing were also suggested by Hu [16], Klein [22] and Gufahr and Nemhauser [12], but were of little practical value because of innumerable computations required.

Ignall [17], in 1965, and Master [27], in 1970, conducted a comparison study of some of the prominent line balancing techniques. Their work is one of the most valuable contributions to the area of line balancing because no attempt had been made in the past to compare the various line balancing techniques. Master [27] considered the models proposed by :

1. Helgeson and Birnie
2. Hoffman
3. Arcus
4. Held, Karp and Shershian
5. Kilbridge and Wester

Besides these, he also considered some of the commonly used heuristics, for generation of sequence of tasks. The heuristics, he considered for the purpose of line balancing are :

1. Lexicographic Order
2. Number of immediate follower task rule
3. Task time ordered rule
4. Random sampling rule

Master's investigations revealed that, the models of Held and Arcus consistently stand out as the best, requiring the least idle time for any number of stations. But Held's technique seems to have limitation on to the size of the problem to which it is applicable, otherwise, as Master claims, it is the best performer. Lexicographic Order Rule was found to be worst throughout. Master concluded that only COMSOAL by Arcus, and, heuristic line balancing technique by Kilbridge and Wester are the most effective techniques for large size practical problems (having more than 40 tasks). For moderate and small size problems, Held's technique produces the best results.

All the methodologies mentioned above do not consider the feature of variability of task time, which is a very important factor in deciding the actual output from the line. The time requirement for each task is generally not constant and may be represented by some emperically determined probability distribution. This

problem was taken up by some authors, namely, Moodie and Young [28], Mansoor [26] and Ramsing and Downing [32].

Moodie and Young tackled the variability feature by alloting some allowances to the operators, so that actual station time does not exceed the designed cycle time, for a given confidence level. Provision of allowance to the operator reduces the output rate from the line and therefore, does not assure to obtain an optimal solution, so long as the objective function concerns with the cost aspect of the line.

Ramsing and Downing [32] also tried to incorporate the variability feature of task times in the manner which was suggested by Moodie and Young. But their treatment is different from that of the latter in the following ways. Ramsing and Downing used sampling technique to determine the task times, for a given confidence level, and then, with the sampled values of the task times, they balanced the line using RPW technique. These authors also, like Moodie and Young, did not consider the possibility of introduction of buffer inventories, as was later suggested by Freeman [9]. As per Freeman and others [2,3] introduction of buffer inventory is more economical way of achieving improved productivity.

Mansoor [26], also introduced another model, which allots allowances to the operators. Along with this,

he has also suggested to introduce an incentive plan for the operators. It might be said that the incentive plans suggested by them, while convenient for their model, seems to be totally impracticable in real life setting.

None of these authors [26, 28, 32] consider interstage inventories, paralleling of stations, failure of stations, etc. The fact is that all the models suggested so far, whether incorporating variability of task times or not, stress more upon reduction of total idle time instead of minimizing total operating costs. The present work is an attempt to develop a cost model of the assembly line, which also considers the following features of the line, to represent it realistically :

1. Variability in the task times,
2. Failure characteristics of the work stations and,
3. A situation where, at least one task is so large in its time contents that it cannot be performed by a single operator to fulfil the requirement of minimum desired output rate.

CHAPTER III

PROBLEM DESCRIPTION AND FORMULATION

3.1 TERMINOLOGY

The following terminology is generally used in literature, in the context of line balancing problems [10] :

1. Task : A feasible subdivision of work content of the total assembly.
2. Task Time : Time content required to perform a task.
3. Work Station : A combination of tasks that must be performed serially forms a work station.
4. Work Station Time : The time requirement for completion of total work load assigned to a work station.
5. Cycle Time : The maximum amount of time available to an operator to produce one unit.
6. Precedence Relationship : The description of the technological order in which the various tasks must be performed in the assembly operation.
7. Line Length : Total number of work stations on an assembly line.
8. Blocking : A station is said to be blocked,

   a. if work on the item is completed but it cannot be passed to the next station either because the next operator is not free, or there is no available space for the inprocess inventory, and

b. if an operator, though free, is unable to serve an item either because of the exhausted inprocess inventory or the preceding operator has not finished the work assigned to him.

9. Buffer Inventory : Semi-finished items stored between the two work stations to lessen blocking. It is also termed as inprocess inventory.

10. Slack Time : This is the idle time at a particular station. It is defined as the difference of the cycle time and the work station time of a station.

11. Balance Delay : It is a measure of the total idle time on the line due to the imperfect division of work among stations.

3.2 GENERAL DESCRIPTION

The problem of line balancing is essentially to distribute the total work load of the assembly line among the operators as uniformly and compactly as possible, keeping in view the desired output rate and the precedence restrictions imposed by the various assembly operations to be performed on the product.

The existing models till to-date, fail to account for the following aspects of an assembly line problem..

They are :

(1) An operator cannot follow, in general, the times specified by the design. The involved human element brings about an uncertainty in the work station times. Hence, the consideration of the work station time as deterministic value is inadequate to represent the system realistically. The work station time should be considered as a stochastic variable following a probability distribution function.

(2) The variability in the time taken by an operator on a station results either in idle time of an operator or a high level of inprocess inventory. Both of these result in a reduced output compared to the value for which the line has been designed. The problem can be solved by introduction of optimum buffer inventories to account for the variabilities in the task times. An infinite level of buffer inventory between each pair of stations will remove the blocking altogether hence increases the output rate to the maximum. But, it is neither economical nor feasible to have an infinite level of buffer inventory. On the other hand, a zero buffer inventory will cause maximum blocking and will reduce the output rate to its minimum. Therefore, the need for obtaining a compromise between the two extremes is felt so as to optimize the given objective function for the assembly line.

There are two factors, which govern the buffer capacity sizes of the line. They are :

1. Variance of work station time, and
2. Buffer holding cost.

The optimization models, reported in the literature, for the assembly line, have assumed equal variances of the station times and equal buffer capacities in between each adjacent pair of stations. In reality, the variances of the work station times need not be equal. The amount of variance at a station depends upon the nature of the work assigned to the work station. Therefore, it should not be taken as equal for all the stations.

Most of the models developed so far, consider cost due to buffer inventory based on a buffer cost value averaged out over the whole line. In a real life situation, the value of the semifinished product increases as it passes through the line and therefore, the holding cost associated, should also be considered in accordance with the change in values of the product. This variation in values, and hence buffer holding costs, becomes more significant for long assembly lines, having sometimes the difference of values, between initial and final stage as high as 100 to 200 times.

(3) The deterministic models of the line balancing also fail to account for the failures of the work stations. The

failure of a work station is also a stochastic variable following a particular probability distribution. The failure of a work station very much affects the output rate of a system. The effect of a work station failure can be nullified by choosing an optimum level of buffer inventory.

(4) Most of the models reported in the literature assume that the task times for all the tasks of the assembly are less than or equal to the cycle time. But, there are cases where there exists at least one task in the assembly, whose task time requirement cannot be met by a single work station within the specified cycle. A general model of line balancing should incorporate this feature.

3.3 THE PROBLEM

Before delving into the problem formulation proper, it is essential to have a better understanding of the system under study, so as to help modelling. Therefore, a description of some of the important features of the system is presented here.

There are m tasks in the whole assembly line. These tasks are to be assigned among the n work stations uniformly and compactly, following the precedence restrictions. The assignment of tasks to the various work stations is done by the technique explained in Phase I of Chapter IV.

The task times, and hence the work station times are random variables, therefore, buffer inventories are introduced to absorb this variability.

A general assembly line can be represented schematically as shown in figure 1.

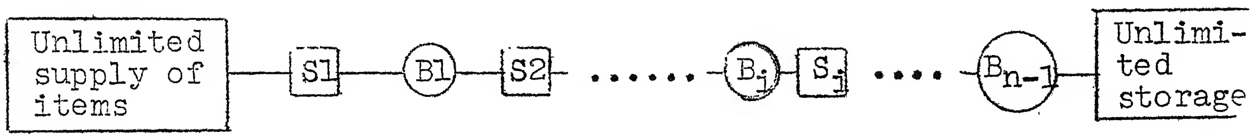


Fig. 1

In figure 1, S_i : indicates the work station i and,

B_i : shows the buffer space between stations S_i and S_{i+1}

S_i in the figure 1 represents the work station i, loaded with the various tasks.

The following assumptions have been made here :

1. There is an unlimited supply of items for the first work station.
2. There is an unlimited storage space following the last work station.

The first work station, S_1, picks up an item from the supply point, performs the operations required on it and then passes it on to the next work station S_2. Similarly, the item passes through every work station, in the prescribed sequence. Every work station adds value to the item and then passes it to the next work station. The process is continued till the finished item comes out of the line.

The assembly line, shown in figure 1, can be considered as a single channel queuing system with n subsystems, each consisting of a service facility and the items in the queue, waiting for the service. The work stations are service facilities subject to the random failure and the buffer inventories are the queues. There are two possible events which change the state of the subsystem :

1. Arrival of an item in the subsystem,
2. Departure of an item from the subsystem after being serviced.

A departure of an item from one subsystem constitutes arrival of an item to next subsystem. The arrival and departure of the items are described as follows :

The work station S_i can pick up an item from B_{i-1} as soon as S_i is free to serve another item after serving last item it had. This is subject to the condition that the work station S_i does not fail in the mean time. If there is no item in B_{i-1} then S_i waits till an item is available to B_{i-1} from S_{i-1} and this process continues. This is valid for all S_i, $i = 2,3,\ldots,n$. This is not valid for S_1 because there is always supply of items. It is ensured that only a single unit will be served at a time by the work station.

The work station S_i can deliver an item to B_i as soon as it completes the service on the item. If B_i is full, the work station S_i waits till an item is removed from B_i by work station S_{i+1} and then delivers the item

to B_i. This is valid for all work stations except the last one, for which unlimited buffer storage is available.

3.4 PROBLEM STATEMENT

The problem of line balancing may be stated as follows :

Given,

1. the total number of tasks to be performed in the assembly line (i.e., the total work load of the line)
2. the precedence relationships among the tasks,
3. the mean and the standard deviation of each task time, and
4. the total number of work stations available to the assembly line.

Then, the problem is to assign the tasks to the work stations of the assembly line, following the precedence restrictions, in order to optimize the objective function (the various objective functions are discussed in the following section). This is to be done accounting for the following features of the assembly line.

1. Variability of the task time,
2. Failures of the work station,
3. A case where cycle time is less than at least one of the task times.

The problem requires following data as the input to the model :

1. Total number of task in the assembly line
2. Mean and standard deviation of each task time
3. The precedence relationship of the tasks
4. Total number of operators available
5. Failure characteristics of the work stations
6. A starting value for the cycle time.

3.5 THE OBJECTIVE FUNCTIONS

The various objective functions related with the cost of the assembly line systems are :

1. Minimization of cost per unit,
2. Maximization of profit per unit,
3. Maximization of profit rate.

Although the management may be interested in the objective functions related with the profit functions of the system, yet it cannot be justified here because of its subjective nature. A distinct estimation of contribution of various productive and nonproductive assembly operations to the total profit earned is very difficult. Therefore, the present work restricts itself to the Ist objective function, namely, minimization of cost per unit produced from the line.

The following chapter is devoted to the methodology to solve the problem stated in section 3.4. The model developed in chapter 4 has been validated in Chapter 5.

CHAPTER IV

DEVELOPMENT OF MODEL AND SOLUTION PROCEDURE

The problem stated in section 3.3 has been divided into two phases :

Phase I : Balance the line using mean task times and determine the maximum output rate that can be achieved from the line. In this phase, the task times are treated as deterministic.

Phase II: Determine the optimal levels of buffer inventories between stations accounting for probabilistic nature of the task times and failure characteristics of all the work stations, which minimizes the total cost of production per unit produced from the line. This phase also determines the expected output rate from the line.

Both the phases have been discussed separately, in the following sections.

4.1 PHASE I

4.1.1 Assumptions :

The following assumptions are made to develop the model for this phase :

1. Only one operator is assigned to each station.
2. No operator is fixed for any particular job.

3. No operator is absent. This means that the management ensures that the needed number of operators are available on the working days.
4. It is a single model assembly line.
5. There are no zoning constraints.
6. The attrition allowance, i.e., the percentage defective is independent of the output rate.
7. The task times are independent of one another.
8. The efficiency of all the work stations is same.

4.1.2 Nomenclature :

1. m : total number of tasks in the assembly
2. n : total number of operators available
3. μ_i : mean task time for i^{th} task, $i = 1,2,\ldots,m$
4. σ_i : standard deviation of i^{th} task time.
5. C_1 : initial cycle time
6. F_{ij} : j^{th} element of i^{th} row of Immediate Follower Matrix F_{ij} indicates the task number, which immediately follows i^{th} task.
7. NT_i: total number of tasks immediately following i^{th} task
8. n' : maximum of NT_i, $i = 1,2,\ldots,m$
9. G_{ij} : j^{th} element of i^{th} row of a binary matrix called 'Follower Matrix'

   $G_{ij} = 1$ implies that j^{th} task immediately follows i^{th} task.

$G_{ij} = 0$ implies that j^{th} task does not immediately follow i^{th} task

10. H_{ij} : j^{th} element of i^{th} row of a binary matrix called 'Total Follower Matrix'
$H_{ij} = 1$ implies that j^{th} task follows i^{th} task
$H_{ij} = 0$ implies that j^{th} task does not follow i^{th} task

11. C_k : Cycle time for k^{th} iteration

12. K_i : i^{th} element of the sequence of tasks generated by the method of line balancing

13. S'_j : Slack time left at j^{th} work station, $j=1,\ldots,n$

14. t_j : number of tasks assigned to j^{th} station

15. d_k : balance delay for k^{th} iteration

16. T_j : Time available with j^{th} station till latest assignment of task there.

17. A_i, B_i and D_i : are arrays as discussed in section 4.1.4.

18. W_{i1}, W_{i2}, W_{i3}, W_{i4} : are weightages as discussed in section 4.1.4

19. R_i : Summation of task times of the tasks following i^{th} task.

Few other notations for the phase are described later in this chapter as the model is developed.

4.1.3 Balancing the Line :

The implication of balancing a line is to assign the various tasks of the assembly line among the various work stations uniformly and compactly while following

the precedence restrictions. Therefore, the best balance is one which simultaneously achieves maximum compactness and maximum uniformity in the allocation of total work load among the work stations. Balance Delay Ratio (BDR), a measure of total idle time of the line, is one of the most commonly used measures to represent the compactness aspect of work load allocation among the work stations. It is defined as follows :

$$BDR = \frac{n \times c_k - \sum_{i=1}^{m} \mu_i}{n \times c_k}$$

where n, c_k, μ_i are as explained in section 4.1.2.

The smaller the value of BDR, the more compact is the assignment. A zero BDR value implies that there is no idle time in the line and the line is perfectly balanced. But for most of the practical problems the minimum achievable BDR is a non-zero quantity because of the descrete values of the task times. This indicates that some work stations are more heavily loaded than the others and that the line contains some idle time. It is desired then, to distribute the total idle time evenly among the work stations and make the work load allotment more uniform.

Almost all the line balancing techniques reported in literature achieve the first objective only i.e., to attain the minimum idle time of the line. Uniform

distribution of the total idle time among the work stations is not been reported so far. This objective can be achieved if several solutions having the same minimum value of the BDR for the balanced line are available. These solutions are then used to select the one which gives the most uniform distribution of the total idle time among the work stations.

The problem handled in Phase I is divided into two parts to achieve the following two objectives :

Objective 1 : To achieve maximum compactness of the allocation of tasks to the various work stations, that is, to achieve minimum BDR.

Objective 2 : Having achieved the balance with minimum BDR, to distribute the total idle time uniformly among the work stations.

4.1.4 Part I :

The aim in this part is to accomplish objective 1 i.e. to achieve minimum BDR. COMSOAL technique developed by Arcus [1] has been chosen to achieve this objective. The reasons for this choice are given below.

1. Master and Others [27, 17] have reported that COMSOAL technique is found to be the best for large size problems (problems having more than 40 tasks). They claim that this requires much less computer time as compared to other techniques e.g., the

Dynamic Programming approach by Held (13). It also yields optimum solution in most of the cases.

2. This technique, unlike other techniques gives more than one solution which have the same value of the BDR. These solutions are then used to achieve the objective 2, as will be discussed later.

The essence of COMSOAL lies in generating the feasible sequences of tasks (i.e. the sequences of the tasks obeying precedence relationships) and to assign them to the various work stations. The sequences giving minimum number of operators are said to form the list of best ones. To achieve the maximum output rate for a fixed number of operators the same procedure is repeated by reducing the cycle time gradually till the required number of operators for a cycle time exceeds the number of operators available. Thus, the sequence giving minimum cycle time for a given line length are the best ones.

Since the generation of every feasible sequence for a large scale problem requires enormous effort Arcus suggested a few heuristic approaches to generate the feasible sequences. This approach reduces the computational effort and at the same time gives optimal solutions in most of the cases. Among the many heuristics suggested the following have been selected in this work which were found to be the best by Arcus.

1. Assign weightages to the tasks in proportion to their mean task times. The weightages are calculated as follows : The weightage for i^{th} candidate task,

$$W_{i1} = \frac{\mu_i}{\sum_{i=1}^{l} \mu_i}$$

where, is the total number of tasks which ate candidates for assignment to the work station being considered presently.

2. The tasks are weighted in proportion to a number 1/X where, X is equal to the total number of unassigned tasks less the number of all the tasks that follow the tasks under consideration less unity. Mathematically, the weightages can be represented as follows :

$$b_i = \frac{1}{(k-f_i-1)}$$

$$W_{i2} = \frac{b_i}{\sum_{i=1}^{l} b_i}$$

where,

f_i : total number of tasks following i^{th} task

k : total number of unassigned tasks at the time when i^{th} task is candidate for assignment.

3. A task is weighted in proportion to a number determined by adding 1 to the total number of tasks following it. Mathematically, it can be expressed

as,

$$W_{i3} = \frac{(f_i + 1)}{\sum_{i=1}^{\ell}(f_i + 1)}$$

4. A task is weighted in proportion to the summation of task times following it and including itself. The expression for the weightage for i^{th} candidate task is,

$$W_{i4} = \frac{(\mu_i + R_i)}{\sum_{i=1}^{\ell}(\mu_i + R_i)}$$

It is to be noted that all the weights are normalized. After all the above mentioned weightages are determined, final weightages W_i are calculated for all the candidate tasks by multiplying different weightages of the same tasks. That is,

$$W_i = W_{i1} \times W_{i2} \times W_{i3} \times W_{i4}. \quad \text{for, } i = 1,2,\ldots,\ell$$

These final weightages W_i are also normalized to obtain an array r_i of such final normalized weightages, for all the candidate tasks ready for the assignment.

4.1.4.1 Generation of feasible sequences for balancing the line -

Following are the steps needed in generation of the feasible sequences of the tasks.

Step 1 : A matrix F_{ij} is formed such that the elements in i^{th} row of this matrix indicate the tasks for which i^{th} task precedes immediately.

Step 2 : With the help of the matrix F_{ij}. form an array A_i. for all the tasks of the assembly line such that i^{th} element of this array represents the total number of tasks immediately preceding i^{th} task.

Step 3 : Form a set B_i containing the tasks from array A_i which have zero value in array A_i.

Step 4 : Scan the tasks in the set B_i satisfying the following condition :

$$\mu_i \leq T_j \text{ for all } i \in B_i$$

Here j is the station being loaded at the time of latest assignment of task. Place all such tasks in the set D_i.

If no such task is found in set B_i for which above condition is satisfied, consider a next work station with the new value of $T_j = c_k$ for the fresh assignments of candidate tasks in set D_i. Here c_k is the cycle time being considered for the current k^{th} iteration. Step 4 is then repeated with the new values.

<u>Step 5</u> : All the elements of set D_i are the candidate tasks for the assignment to the work station under consideration. All the tasks in the set D_i are now weighted according to the procedure described in section 4.1.4. The selection of task from set D_i is biased in accordance with the weightages assigned to the tasks and is not purely random. The g^{th} task selected in this manner is allocated to the j^{th} work station. This allocation results in the reduction of the time available to this station by an amount μ_q. That is, time available at i^{th} station for other allocations is now $(T_j - \mu_q)$. The number of tasks t_j assigned to station j is also incremented by unity.

<u>Step 6</u> : After the q^{th} task is assigned to station j in Step 5 the values of A_i are modified as follows :

A_q = a high value, say 10,000

$A_j = A_j - 1 \ \forall_j \in F_{qj}$

$A_j = A_j \forall_j \in F_{gj}$ and $j \neq q$

Step 7 : The steps 1 to 6 are repeated to generate a sequence of tasks till all the tasks are assigned to the work stations.

A number of such sequences are generated by repeating the above procedure. The sequences yielding minimum number of work stations are selected.

The procedure described so far does not include the case when $\mu_i > c_k$ for at least one i. The following section describes the procedure to account for such a case.

4.1.4.2 Consideration of a case where $\mu_i > c_k$ for at least one i .

As discussed in section 3.2 a general model of line balancing should indlude this feature. To take this case into account, the procedure adopted in this work is to introduce parallel stations. Since the task r for which $\mu_r > c_k$ cannot be further subdivided into smaller tasks, the station time of each of the parallel stations must be greater than the value μ_r. In fact, all the parallel stations according to the procedure adopted here will have work station times as the integer multiples of the cycle time such that the task r can be performed within this higher value of work-station times for the parallel stations. All the work-stations are

considered exactly identical as far as the nature of work load assigned to them is considered. If all the parallel stations are taken together and are considered as a single equivalent work station the effective cycle time for this unit will be equal to cycle time for the rest of the stations.

After assigning task r to the parallel stations, if any time is left with them, it may be used to allocate more tasks to the stations, if the precedence restrictions permit to do so. The details of the logic used for this case are given below.

The number of parallel stations required to perform ~~the~~ task r

$$p_r = \frac{\mu_r}{c_k} + f$$

where, f is a fraction $(0 < f < 1)$ added to make p_r an integer.

Thus, effective time available with all the parallel stations = $p_r \times c_k$. After assigning task r to parallel stations the time available with each of the parallel station for assignment of other tasks is,

$$p_r' = p_r \times c_k - \mu_r$$

For the task r a new term 'Virtual Task Time' is defined here as follows :

$$\text{Virtual Task Time } \mu'_r = c - p'_r$$

It is to be noted that μ'_r is not the actual task time for task r. The balancing is then carried out according to the procedure discussed in previous section, with the difference that μ_r is now replaced by μ'_r and n is replaced by a new value of $n = (n-p_k + 1)$. When the final balance is achieved, the station having task r is replaced by p_r number of similar parallel stations, loaded identically.

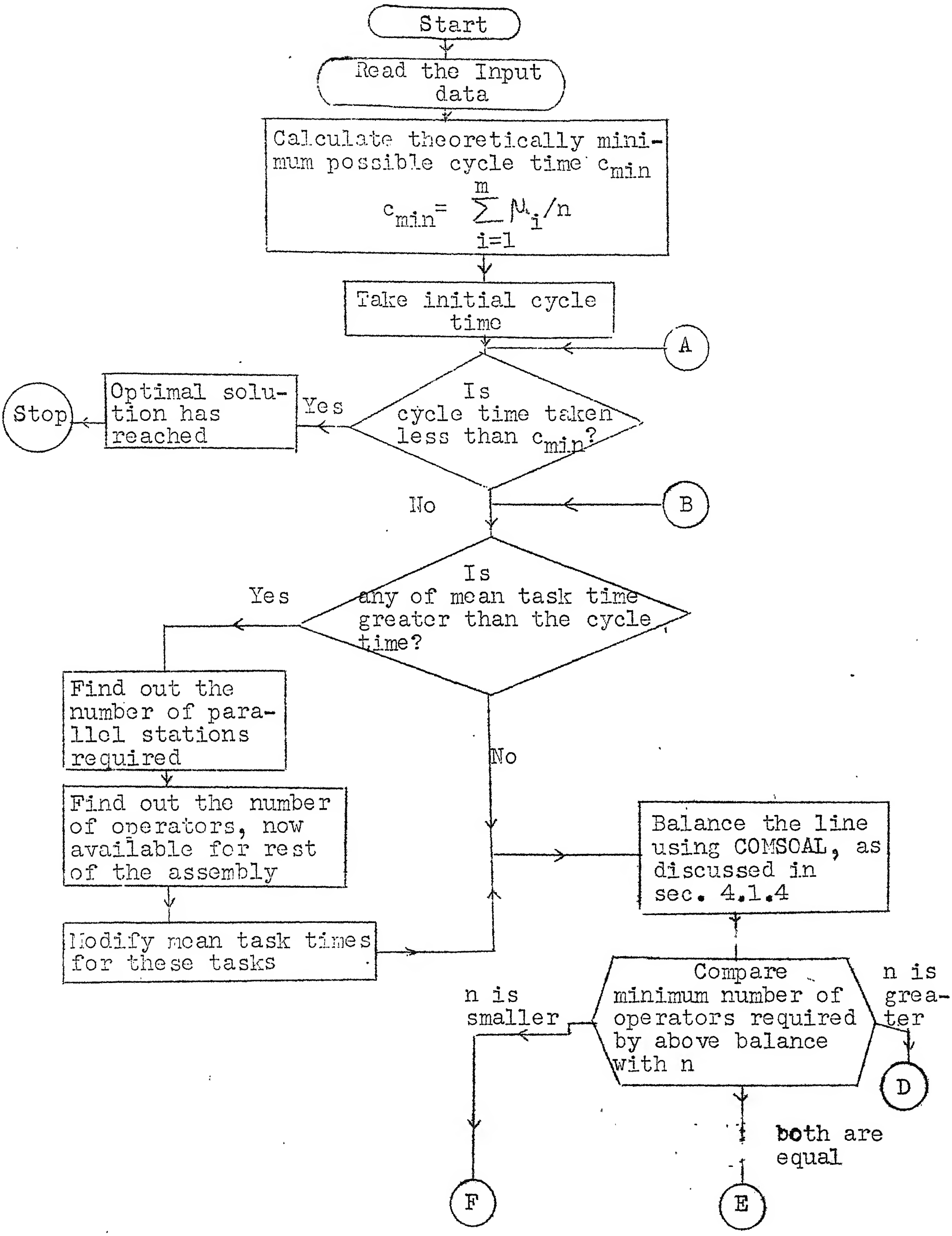
The flow chart in figure 2 shows in logical form the procedure developed for attaining the first objective. The next section explains the procedure for accomplishing the second objective.

4.1.5 Part II :

In this part the task sequences obtained in part I are examined again with a view to uniformly distribute the total idle time of the line among the work stations.

To evaluate the uniformity in distribution of the total idle time among the various work stations, an index called the "Smoothness Index" has been introduced. Smoothness Index is defined as,

$$S.I. = \sqrt{\sum_{i=1}^{m} (S'_{av} - S'_i)^2}$$


Start
Read the Input data
Calculate theoretically minimum possible cycle time c_{min}
$c_{min} = \sum_{i=1}^{m} \mu_i / n$
Take initial cycle time
A
Is cycle time taken less than c_{min}?
Yes
Optimal solution has reached
Stop
No
B
Is any of mean task time greater than the cycle time?
Yes
Find out the number of parallel stations required
Find out the number of operators, now available for rest of the assembly
Modify mean task times for these tasks
No
Balance the line using COMSOAL, as discussed in sec. 4.1.4
Compare minimum number of operators required by above balance with n
n is smaller
n is greater
both are equal
D
E
F

(CONTD...)

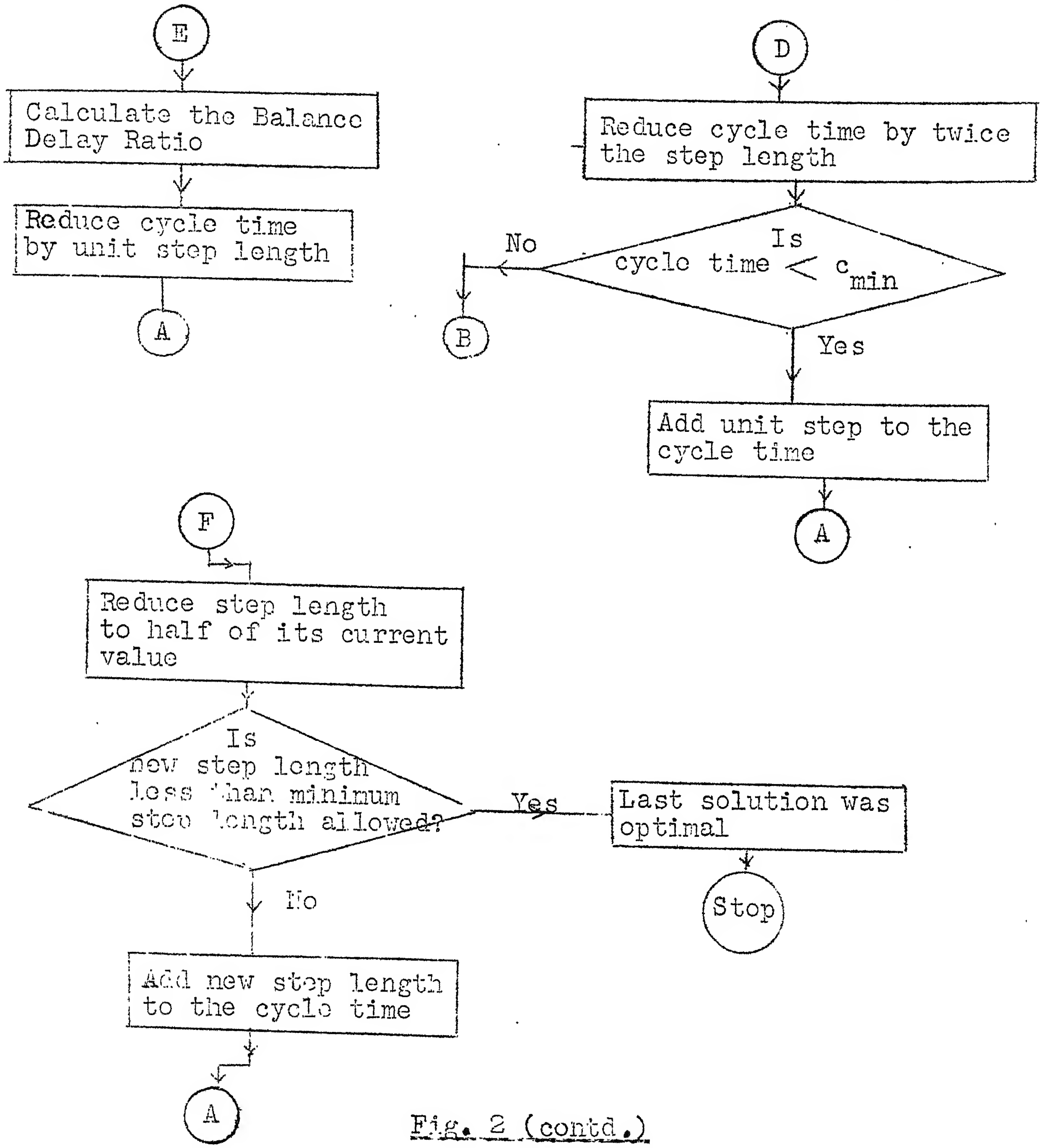

E
Calculate the Balance Delay Ratio
Reduce cycle time by unit step length
A
D
Reduce cycle time by twice the step length
Is cycle time < c_{min}
No
B
Yes
Add unit step to the cycle time
A
F
Reduce step length to half of its current value
Is new step length less than minimum step length allowed?
Yes
Last solution was optimal
Stop
No
Add new step length to the cycle time
A

Fig. 2 (contd.)

where, S.I. = Smoothness Index

S_i' = Slack time at i^{th} work station

S_{av}' = the average of the slack times of all the work stations

S.I. is a measure of the distribution of the total idle time of the line among the work stations. The interpretation is that for the task sequence having the minimum value of S.I., the dispersion of the slack times of the work station about its mean value is minimum. This suggests that the slack times are distributed most uniformly among the work stations. In fact, the smaller the S.I. more uniform is the distribution of the idle time.

The task sequences having S.I. values close to the minimum value of S.I. may be treated as equally good sequences, and thus provide the line supervisor with alternatives. The line supervisor can then select one of these alternatives according to his own convenience.

It should be noted that the definition of S.I. as given above is different from the one proposed by Moodie and Young [28]. They have defined S.I. as,

$$S.I. = \sqrt{\sum_{i=1}^{m} (S_{max} - S_i)^2}$$

where, S_i - station time of i^{th} work station

S_{max} - maximum station time in the line

They have used the index in a different context. They have employed it for trading and transferring tasks from one work station to another for the purpose of making work station time nearly equal. They have not considered the compactness aspect of the allocation of the tasks to the various work stations.

The procedure of balancing the line, as explained in Phase I has been computerized. Every care has been taken so that it can be applied to any general line balancing problem. The listing of the computer program is given in Appendix B. The comment cards have been put liberally to explain the working of the program and meaning of various important variables used in the program.

Phase I is followed by Phase II, which is explained in the next section.

4.2 PHASE II :

Phase I yielded a balanced line, where the work stations are loaded with tasks uniformly and compactly. This was done using the mean values of the task times. However, in real life situations one comes across

1. Variability in the task times, and
2. Failure of the work stations.

The objective in this phase is to account for the above mentioned two aspects by incorporating optimum buffers between the work stations. The values for buffer capacities are determined with a view to minimize the total

cost of production per unit produced from the line.

An assembly line represented schematically in figure 1 can be considered as a single channel queuing system with n subsystems each consisting of a service facility (work station) and the item (the buffer inventory) in the queue waiting for the service. The service time and service faility's failure are random phenomenon and have some probability distributions associated with them.

The assumptions listed below have been made to develop the model. The basis of these assumptions is the characteristics of the various types of data collected for the assembly line of J.K. Electronics. The details about the data collection will be discussed in Chapter 5. Minor modifications in the methodology can incorporate different probabilistic characteristics of the data. The assumptions made are as follows :

1. The time between two failures of a work station is a negative exponentially distributed random variable.
2. The repair time for each station is a normally distributed random variable.
3. The service time of the work station is a normally distributed stochastic variable.

The mean and the standard deviation of the service time of a work station are given by the following relationships :

$$\mu_{si} = \sum_{j \in k_i} \mu_j \quad \forall i = 1,2,\ldots,n$$

where,

μ_{si} : mean of service time for i^{th} work-station (also called work-station time)

μ_j : mean of the task time for j^{th} task

k_i : set of tasks which are assigned to i^{th} work station in Phase I

$$\sigma_{si} = \sqrt{\sum_{j \in k_i} \sigma_j^2} \qquad i = 1,2,\ldots,n$$

where,

σ_{si} : standard deviation of i^{th} work station time

σ_j: standard deviation of the task time for j^{th} task

It is to be noted here that the task times are normally distributed variables. Also to be noted is that in the expression for standard deviation of the work station time, the correlation among the task times of set k_i has been neglected.

The transactions in the assembly line being considered were explained in section 3.3. This assembly line with the features listed about namely the variability of the task times and the failure of the work stations cannot be solved analytically because of the complexities involved. Therefore, digital simulation has been resorted to solve the problem. A logical model is developed in the present work to represent the above mentioned assembly line system

to be solved on a digital computer.

4.2.1 Terminology :

Input data to the system :

1. n : total number of work-stations in the assembly
2. μ_{si} : mean work-station time for i^{th} work station
3. σ_{si} : standard deviation of i^{th} work-station time
4. λ_{fi} : parameter of negative exponentially distributed interfailure time of i^{th} work station
5. μ_{ri}, σ_{ri} : mean and standard deviation of repair time for i^{th} work-station
6. g_i : is the number of stations in the i^{th} group of V_i
7. u_{ij} : is the i^{th} member station of j^{th} group of V_i
8. m^1 : is the number of items produced after which output rate is computed

Other important variables :

9. a_i : indicates the value of time of the next arrival in i^{th} work station
10. d_i : indicates the value of time of the next departure from the i^{th} work station
11. f_i : indicates the value of time at which i^{th} work station fails
12. q_i : indicates the value equal to the length of queue after i^{th} work station (i.e., number of items in buffer i)

13. ST_i : indicates the status of i^{th} work station

$ST_i = 0$ implies that the station i does not have any item under process and it is ready to take one item from the buffer.

$ST_i=1$ implies that station i is now occupied by an item. It will deliver the item as soon as the service is complete.

14. T : has the value of time of the most recent change in the status of the stations. It is some sort of clock time which gives an indication of the time the process has already taken place.

15. P_i : denotes the number of items produced by i^{th} work station till T.

16. Q_{ij} : denotes the units of time for which i^{th} buffer space has (j-1) items in it

for $i = 1,2,\ldots,(n-1)$ and

$j = 1,2,\ldots,(B_i+ 1)$. Here B_i is the buffer capacity for i^{th} buffer space.

17. E_i : actual time taken by i^{th} work station to produce an item.

18. y_i : has the value of the time of the previous arrival in the i^{th} work station.

19. z_i : has the value of the time of the previous departure from the i^{th} work station.

Following are the design variables of the system :

20. B_i' : contains the value of buffer capacity for i^{th} group of V_i as explained in section 4.2.2.

21. D_i : buffer capacity between i^{th} and $(i+1)^{th}$ work stations.

The other variables and parameters related with cost model have been described in section 4.2.6.

4.2.2 Method for reduction of combinatorial size of the Problem :

As explained in section 3.2, the consideration of different buffer capacity sizes at different work stations makes the problem highly combinatorial and therefore practically unsolvable. A methodology has been developed in this research work to unriddle this problem. The methodology makes possible the allocation of different buffer levels at different buffer spaces.

Consider the assembly line shown in figure 1 of chapter 3. The buffer capacity B_i is dependent upon the variances of all the stations preceding and following it. However, a buffer is mostly effected by the stations adjacent to it on either side. Further, any contributions due to other stations is small and therefore can be neglected. The magnitude of their effect is further

reduced due to buffer allotted between them. Therefore, the buffer capacity is mainly dependent upon the variances of stations S_i and S_{i+1}. Following this assumption, associated with each B_i is a number V_i to help in the development of the model. The number V_i is defined as follows :

V_i = Variance of station S_i + variance of station S_{i+1}

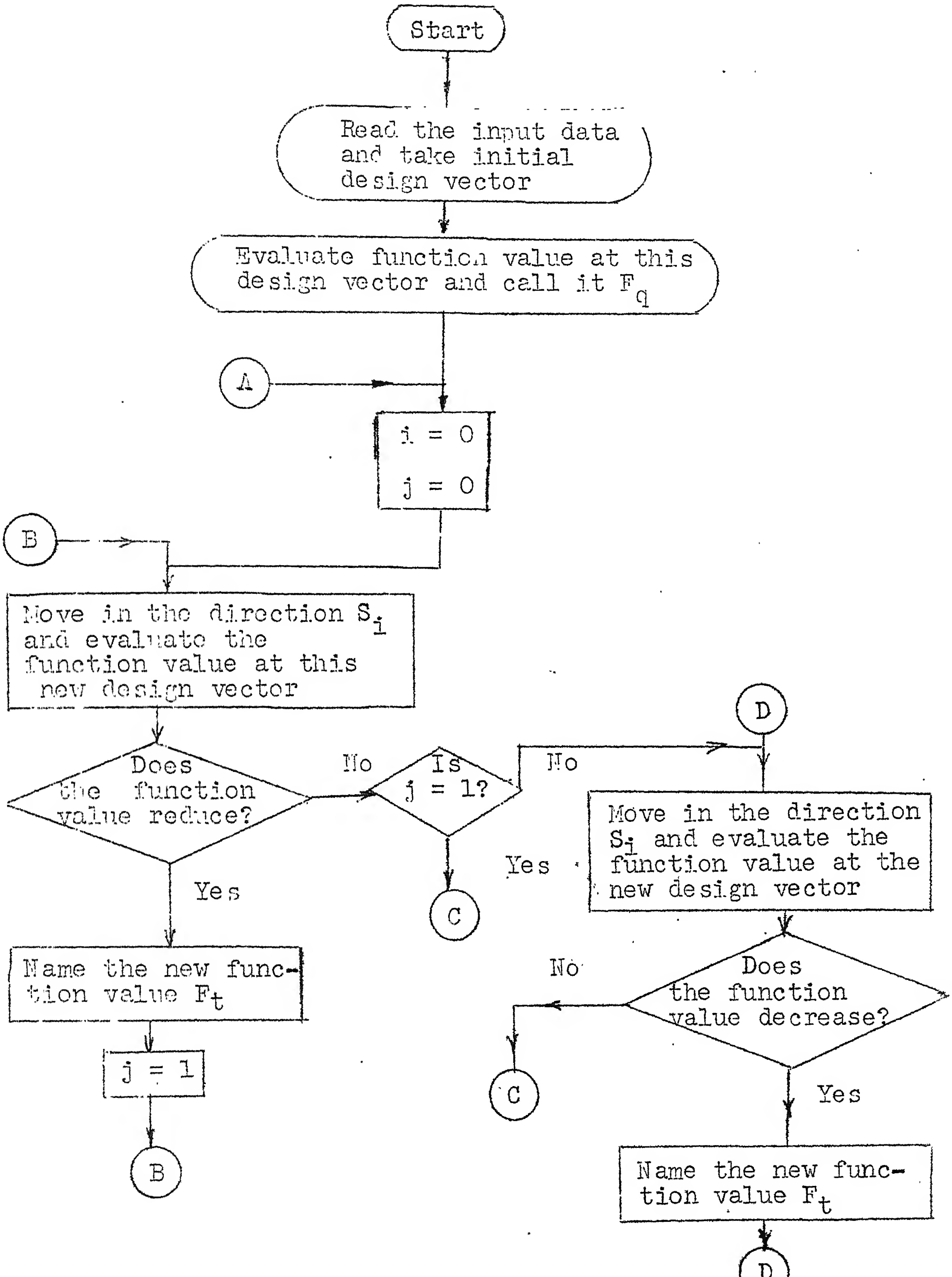
The following scheme is followed to reduce the combinatorial size of the problem. The steps are given below.

1. Arrange all the B_i $i = 1,2,\ldots,n-1$ in the increasing order of the values of V_i
2. Divide the whole range of V_i into k ($k < n-1$) subdivisions such that each subdivision have equal number of buffer spaces. If it is not possible to have equal number of buffer spaces in each of k groups then the buffer say B_j is included in the group to which V_j is nearest.
3. Assign same initial values of B_i to all i belonging to group j. Call these values B_j', $j = 1,2,\ldots,k$. The values B_j' may be different for each j, $j = 1,2,\ldots,k$. It is also to be noted that during the simulation of the assembly line, whenever the level of buffer B_j' is changed for a particular group j, all the B_i contained in j^{th} group are changed accordingly.

4.2.3 Optimization Procedure :

After following the scheme explained in previous section, there are k ($k < n-1$) design variables, namely, the values of B_i' for k groups of V_i. The values of B_i' range from zero to infinity and therefore it is not feasible to calculate objective function for each possible combination of their values. In order to determine optimum values of B_i' , $i = 1,2,\ldots k$, and also to reduce search Univariate Search technique [9] is used. The reason for this choice is mainly to have economy in computer time. Most of the optimization techniques other than Univariate method usually require local exploration at intermediate stages of process; e.g., evaluation of the gradient of the function which need many function evaluations. In the present problem each function evaluation requires one simulation run of the process and hence a large computer time is consumed. The Univariate technique on the other hand, does not need this information and therefore, less computer time is required. Another reason for this choice is its simplicity and ability to give good solution for a problem with small number of variables.

Fig. 3 shows the procedure for univariate search schematically with the help of a flow diagram. Following are the steps involved in this procedure.



<u>Fig. 3</u> (Contd...)

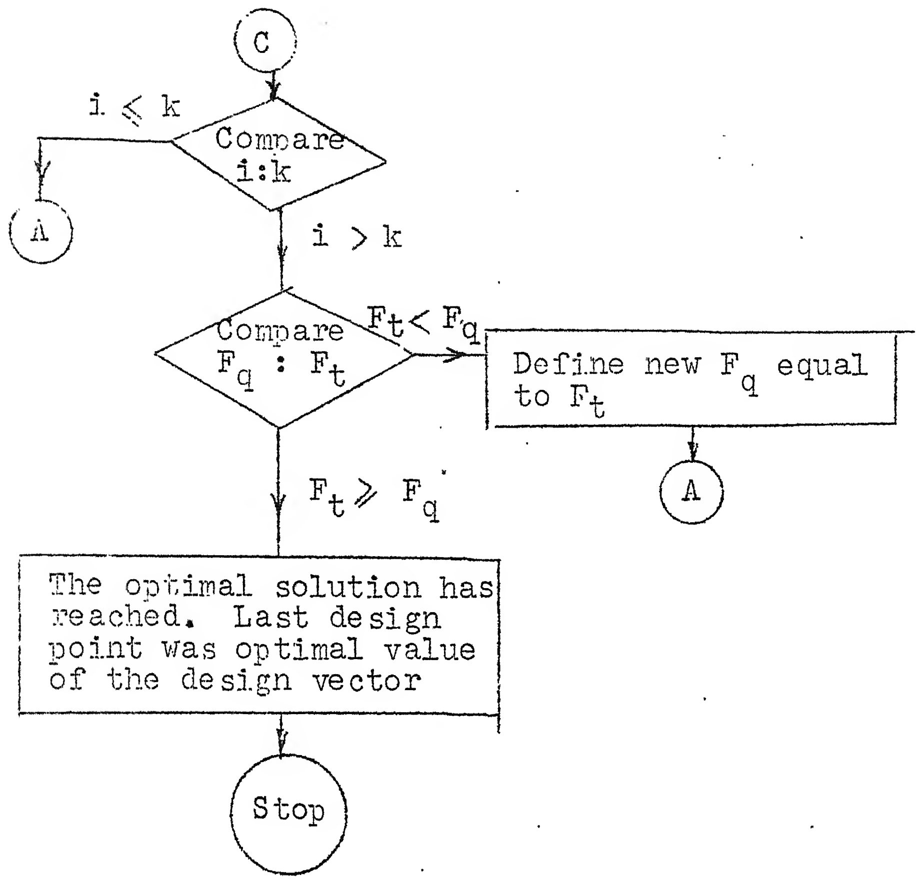

C
i ⩽ k
Compare i:k
A
i > k
Compare F_q : F_t
$F_t < F_q$
Define new F_q equal to F_t
A
$F_t \geqslant F_q$
The optimal solution has reached. Last design point was optimal value of the design vector
Stop

Fig. 3 (Contd.)

1. Choose a starting vector B'_q , $q = 1,2,\ldots,k$
2. Determine whether the predetermined step length t should be positive or negative. In other words, for a particular direction of movement, S_q, does the function decrease in the positive or negative direction of S_q ?
3. Take a step of $\pm$ t in the direction S_q (whichever step length minimizes the function value) and calculate the function value F_t at the new design point $B'_{q+1} = B'_q \pm tS_q$. If the value of F_t at the new point B'_{q+1} is less than the value F_q, increase the step length t by a factor c, where $c > 1$, and repeat step 3 and 4. Otherwise set the design point $B'_{q+1} = B'_q \pm t_p S_q$, where t_p is the last 'successful value' of the step length. Choose the next S_q and repeat steps 2, 3 and 4 for the new direction.
4. Stop when no direction S_q, produces an improvement in the function.

The directions of movement, S_q's, are chosen as a cyclic ordering of the unit vectors e.g., $S_1 = (1,0,0,\ldots,0)$, $S_2 = (0,1,0,\ldots,0),\ldots, S_k = (0,0,\ldots,0,1)$, $S_{k+1}=(1,0,0,\ldots,0)$.

It is to be noted that this procedure gives only the local minima of the function. Therefore, it is preferred to evaluate local minimas from different starting points and select the minimum of all these minimas.

4.2.4 Simulation of the assembly line :

A logical model of an assembly line which represents both of the following features, namely,

1. The variability of the task times, and

2. The failure of the work stations,

has been developed in this research work. Fig. 4, 4(A) and 4(B) show the logical flow diagram for the proposed methodology.

4.2.5. The Steady State of the Assembly line :

During the simulation run the assembly line is assumed to have reached the steady state when the rate of production from the line is stablized. The rate of production is computed in consecutive periods of time in which the line produces m' additional products. That is, the rate of production is computed when the line has produced the $(m')^{th}$, $(2m')^{th}$, $(3m')^{th}$.... product etc. The value of m' is selected arbitrarily and is kept close to the average number of products produced in one shift. The rate of production per unit time is defined as follows:

$$O_{i+1} = \frac{m'}{\text{Time required to produce } (i+1) \times m' \text{ products } - \text{ time required to produce } i \times m' \text{ products}}$$

where, O_{i+1} = rate of production in i^{th} period

m' : a number as discussed above.

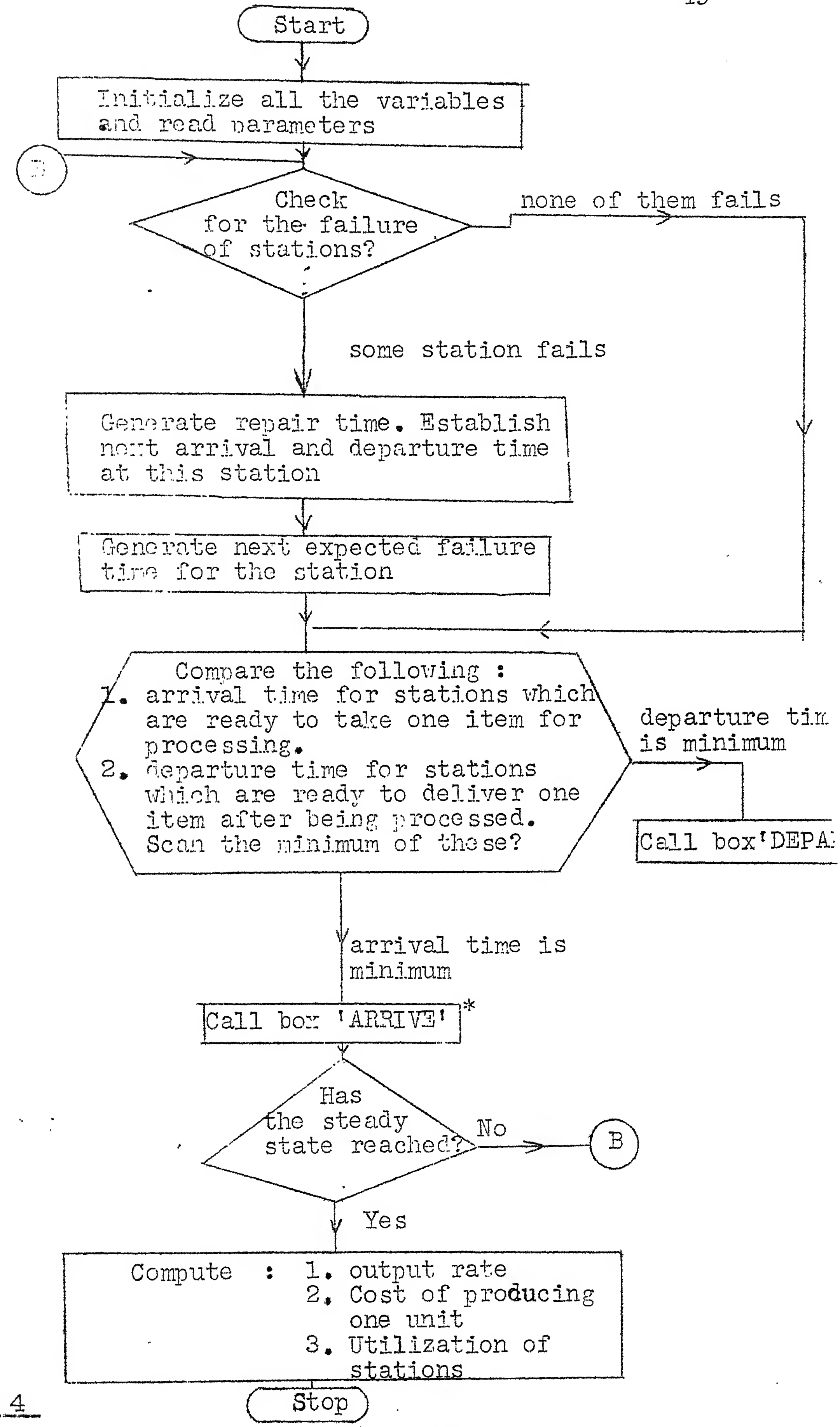


Figure 4

*These boxes are explained in Fig.4(A) and 4(B).

Box 'DEPART'

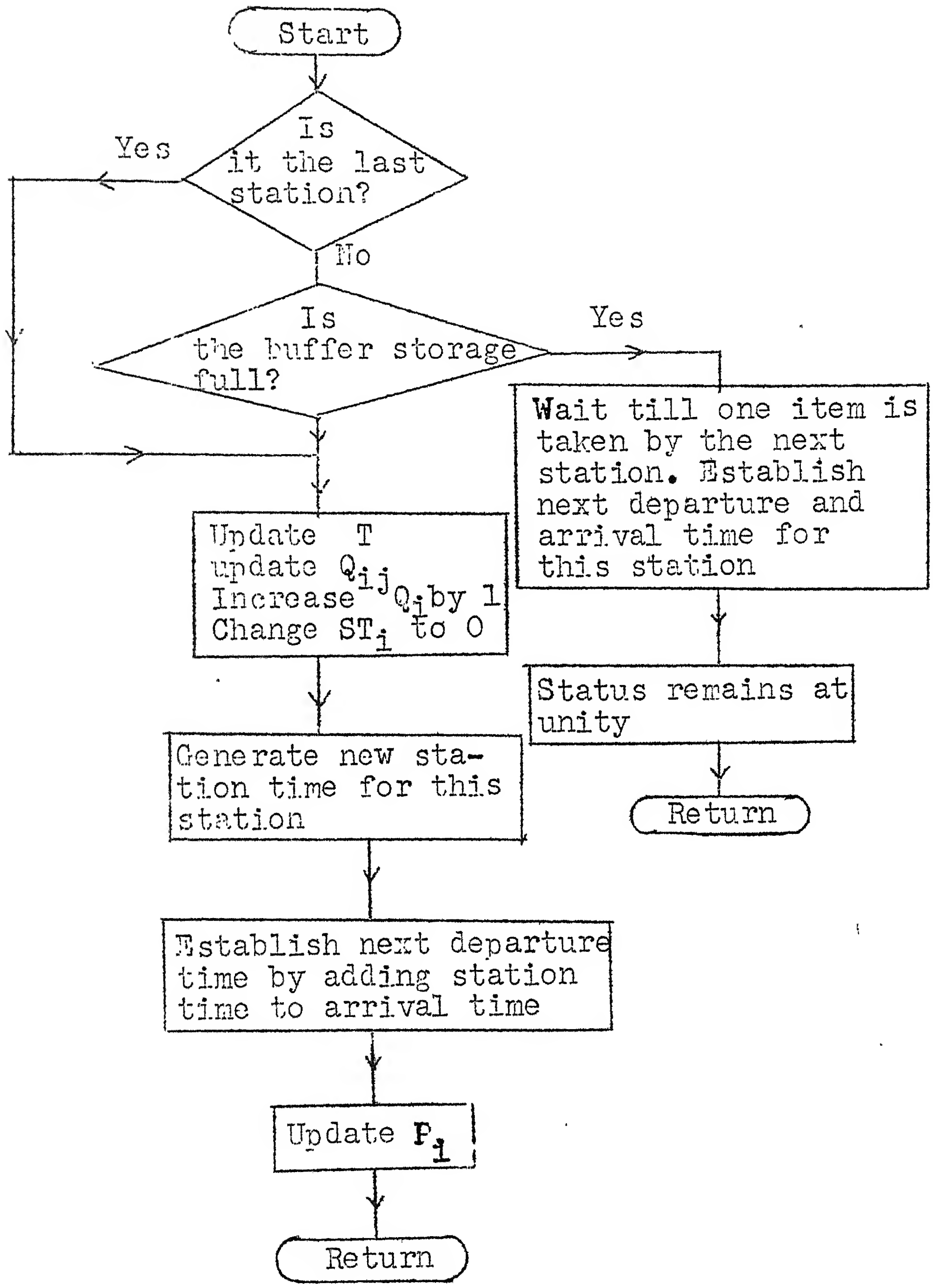
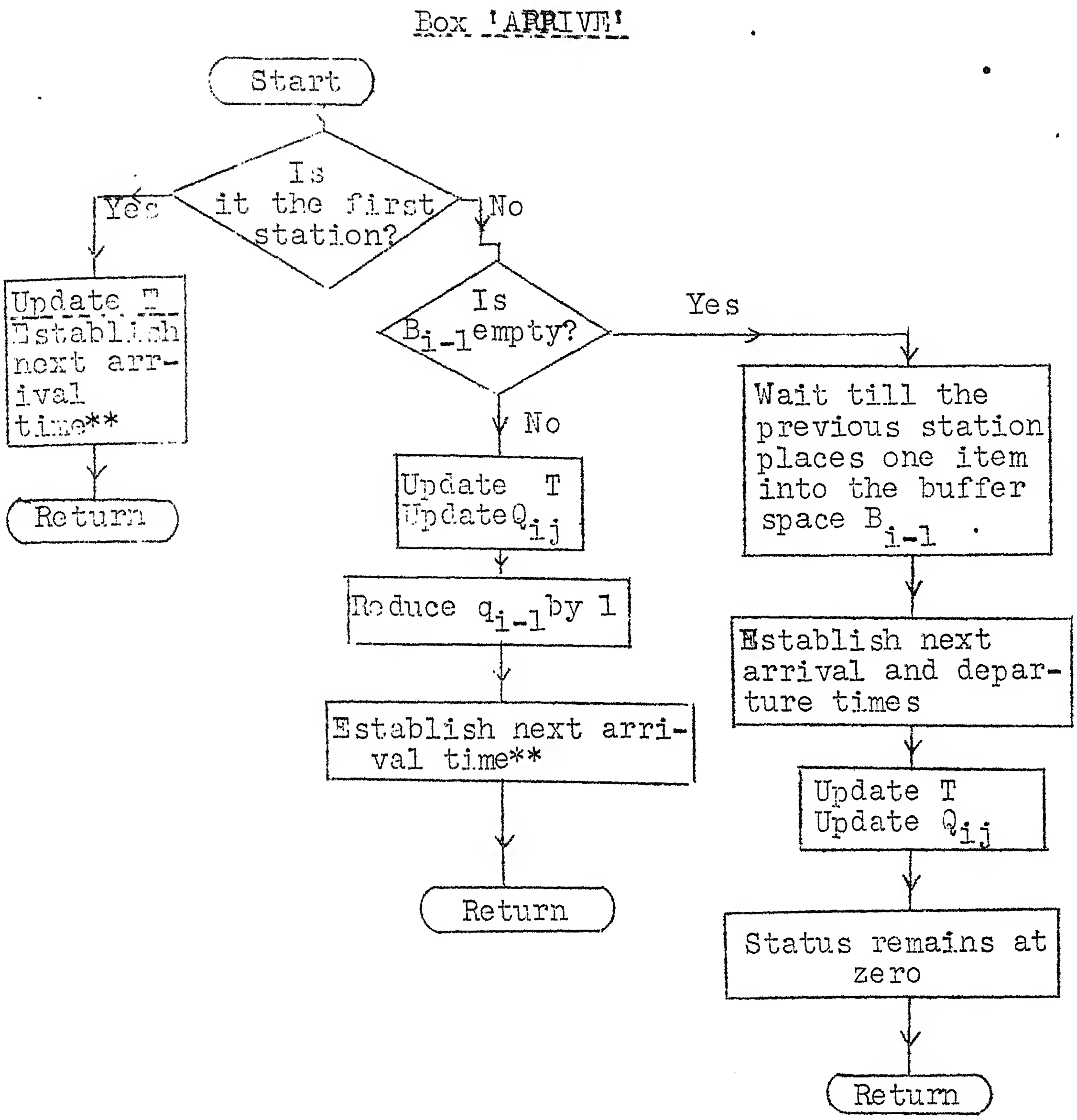


Figure 4(A)



** The stations can pick up an unit immediately after it has finished working on one. Therefore, next arrival times will be equal to current departure times.

Figure 4(B)

Since the transactions in the assembly line are probabilistic in nature, the chances are that the computed values of O_i and O_{i+1} are close to each other even when the line has not reached the steady state. This is depicted clearly in figure 5.

The following procedure has been adopted in this work to avoid above mentioned situation.

1. Let the process continue till p periods. Here p is the number of periods after which the assembly line is expected to reach the steady state. Thus number is estimated by taking a trial simulation run of the system.
2. Calculate m_o-periods moving averages of the rate of production in the p^{th} and $(p-1)^{th}$ period, as given below.

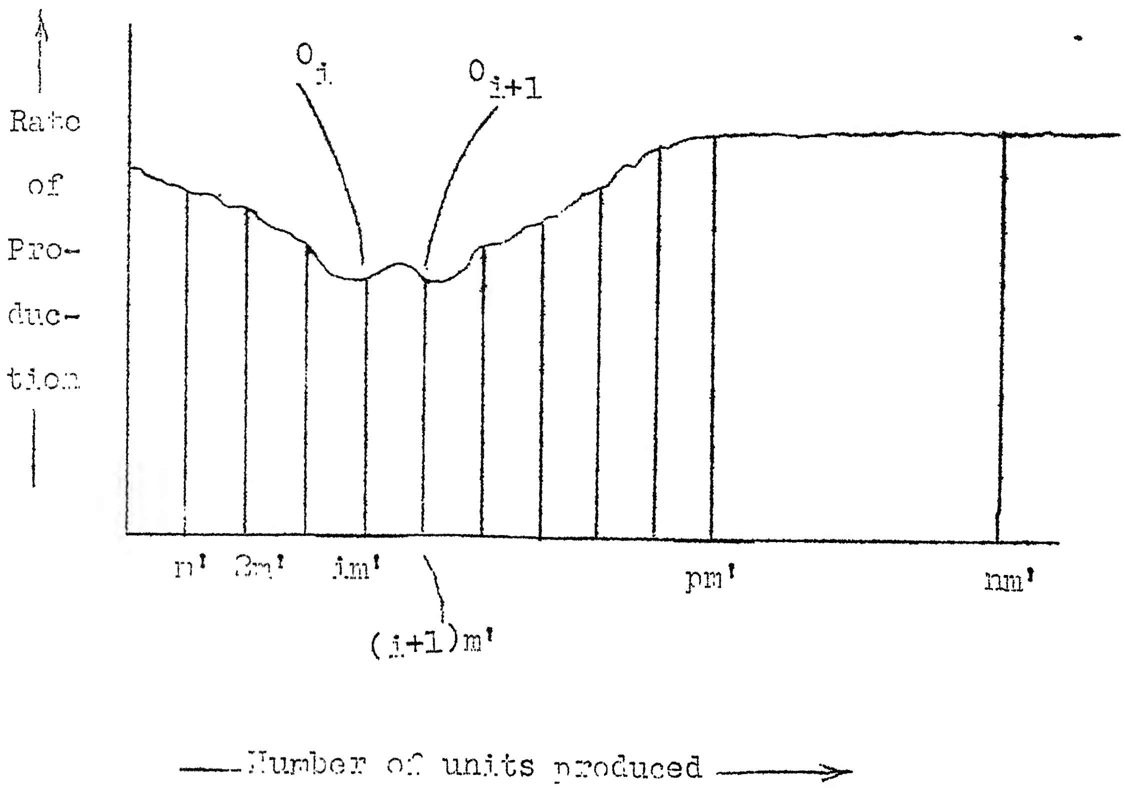
$$A_p = \frac{1}{m_o} \sum_{j=1}^{m_o} O_{p-j}$$

$$A_{p-1} = \frac{1}{m_o} \sum_{j=0}^{m_o-1} O_{p-j}$$

where, A_p = moving average in p^{th} period

A_{p-1} = moving average in $(p-1)^{th}$ period

m_o = an arbitrarily chosen number such that $1 < m_o < p$.



Rate of Pro- duc- tion
O_i
O_{i+1}
m'
$2m'$
im'
$(i+1)m'$
pm'
nm'
Number of units produced

Fig. 5

3. Compare A_p and A_{p-1}. If their difference is less than ϵ (a small quantity, depending upon the accuracy desired) the steady state of the process has been reached. Otherwise increase p by 1, simulate the process for the next period and repeat step 2.

Taking m_o-periods moving average smooths out the randomness of the values of O_i and therefore gives a better indication of the steady state.

4.2.6. Stochastic Convergence of the Rate of Production from the Line :

A test for stochastic convergence is conducted after the assembly line system reaches the steady state. For this purpose the sample size is increased by increasing the total length of simulation run. The sample size is calculated for the desired level of significance. The details of the procedure are given in Chapter 5.

4.2.7. Variance Reduction in the Measurement of the Production Rate :

Following measures have been taken to reduce the random errors in the measurement of the production rate.

1. The use of same set of random numbers for different simulation runs leads to reduced random error for comparison of the stochastic variates. The values of the stochastic variates obtained by using same set of random

numbers are likely to be positively correlated and therefore the measurement of the difference of the stochastic variates will have reduced random errors.

Based upon the above principle, the same set of random numbers have been used for simulation runs for different design points in the optimization procedure.

2. The precision of a given sample is improved by using the concept of Antithetic Rule [34]. According to this rule, two simulation runs, instead of just one which is done usually, are taken. The simulation runs are identical, apart from the fact that the random number sequences used therein are different in both the runs. If $x_1, x_2, \dots, x_n$ is the sequence of random numbers used in the first run, the set to be used in the second run will be $(1-x_1), (1-x_2), \dots (1-x_n)$. The justification is that these two sets of random numbers are strongly negatively correlated and therefore, tend to produce results on the opposite sides of the population mean, and when taken together, will give results closer to the population mean than would be likely otherwise.

4.2.8. The Cost Model :

The following costs have been included into the total cost of production :

1. Fixed labour cost
2. Incentive labour cost
3. Buffer holding cost
4. Fixed overhead cost
5. Variable overhead cost
6. Material cost

1. Fixed Labour Cost :

Total fixed labour cost/shift, $L_c = n \times x_1$

where, n : total number of operators in the line

x_1 : labour wage per labour per shift.

2. Incentive Labour Cost :

Total incentive cost per shift, $I_c = (O_s - O_m) \times x_a \times n$ if $O_s > O_m$

$= 0$ otherwise.

Here, O_s : steady state output rate per shift

O_m : minimum output rate, over which the incentive is paid

x_a : incentive cost of producing an additional item.

3. Buffer holding Cost :

Total buffer holding cost per shift, $H_c = \sum_{i=2}^{n} \bar{Q}_i \times x_{hi}$

where, $\bar{Q}_i$: average buffer inventory level at i^{th} buffer space

x_{hi} : Cost of holding buffer at i^{th} buffer space/shift

4. Fixed Overhead Cost per shift = F_c

5. Variable Overhead Cost :

Total variable overhead cost per shift, $OV = J \times (I_c + L_c)$

where J is a specified factor relating overhead cost with total labour cost.

6. Material Cost :

Total material cost per shift, $M_c = O_s \times x_m$

where, x_m is the material cost per unit

The total cost of production per unit produced from the line is obtained by summing the six costs listed above. Mathematically, the total cost of production per unit is expressed as,

$$CP = \frac{1}{O_s}\left[L_c + I_c + H_c + F_c + OV + M_c\right]$$

The other important parameters to be computed during simulation are given on next page.

1. Percentage Utilization of the work station.

Percent utilization of the work station means the percent of the total time the operator is busy. It is defined as follows :

$$PU_i = \frac{U_i}{T}$$

where PU_i : percent utilization of the i^{th} work station

U_i : total productive time of the i^{th} work station.

T : total clock time of simulation run.

2. Average buffer inventory level at i^{th} buffer space.

This is analogous to the average queue length in a queuing system and is defined as follows :

$$\bar{Q}_i = \frac{\sum_{j=1}^{B_i} j \times Q_{ij}}{T}$$

where $\bar{Q}_i$: average buffer inventory level at i^{th} buffer space

B_i : buffer capacity of i^{th} buffer space

Q_{ij}: as defined in section 4.2.1

The methodology developed in this phase has also been computerized. The listing of the program is given in Appendix B. By making minor changes in the computer program one can incorporate probability distribution for the inter-failure and repair times, other than the ones taken in this work. The next chapter discusses the data acquisition and the validation of the model.

CHAPTER V

DATA ACQUISITION AND ANALYSIS

The model developed in Chapter IV has been tested on an assembly line of J.K. Electronics, Kanpur, one of the leading TV set manufacturers in India.

The assembly line studied is called the 'VIF-SIF' assembly line. This is the main subassembly line of J.K. Electronics. The output from this line, VIF-SIF plate, is one of the major components of a television set. The line consists of six work stations and each work station is manned by an operator. The management wants to retain the number of operators as six and wants to improve the productivity of the line by using scientific planning methods.

In the VIF-SIF assembly line each operator is supposed to perform a prescribed set of tasks. Due to the presence of human element, the time contents of each task are probabilistic in nature. Therefore, the time needed for a set of tasks to be performed on a work station is also probabilistic in nature. A time study was conducted to determine the exact nature of the distribution for the task times. As will be seen later in this chapter, the task times are normally distributed. Using

the Central Limit Theorem*, one can say that the work station time which essentially comprises of various normally distributed task times for the tasks assigned to the work station, will also follow a normal distribution.

Another important feature of this line is that the operators working on the line occasionally go out of the assembly line because of their personal needs e.g., going to the toilet, etc. Since during this period of time there will be no output from the work station, this period has been considered as the failure period of the work station. The other failures e.g., power failure etc. are not considered. This is because these failures, sort of, freeze the process of assembly line altogether. They stop the working of the whole of the assembly line and therefore, neither they affect the general nature of the flow of the items in the line, nor they affect the different levels of the buffers.

The phenomenon of failure of work stations and the repair time (i.e., the time duration after which the operator comes back to his position and starts working again) are also found to be probabilistic in nature. As will be seen later in this chapter, the interfailure time of a work station follows a negative exponential distribution

---

* The Central Limit Theorem states that if x_i are continuous and independent random variables with mean value μ_i and standard deviation value σ_i, then the sum $x = x_1+x_2+\ldots+x_k$ will approach a normal distribution with mean value $\mu = \mu_1 + \mu_2 + \ldots. + \mu_k$ and standard deviation value $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots. + \sigma_k^2}$.

and the repair time of the work station follows a normal distribution. The details of this will be discussed in the section 5.2.

The next section describes the total work load in the VIF-SIF assembly line.

5.1 THE TOTAL WORK LOAD IN VIF-SIF ASSEMBLY LINE

The current procedure adopted for assembling the VIF-SIF plate involves the following steps for each of the operators :

1. Pick up a fresh plate from the buffer,
2. Fix a known number of the components (as specified by the management) onto the plate,
3. Flip the plate,
4. Bring the soldering rod to the plate,
5. Solder all the terminals of the components fixed in step 2,
6. Put the soldering rod back to its place,
7. Bring the pliers to the plate,
8. Trim the terminals of the components soldered in step 5,
9. Put the pliers back to its place,
10. Flip the plate again while placing it in the next buffer after completion of the job.

In the above list steps 1, 3, 4, 6, 7, 9 and 10 form the tasks which are exactly identical for all the work stations and are performed once in each cycle. The tasks in steps 2, 5 and 8 are different for different work stations and are discussed below.

Step 2 comprises of several tasks where each task consists of fixing a component, or a set of components, onto the plate. The tasks of fixing the components onto the plate follow a known precedence restrictions because of the technological constraints. Fig. 6 depicts these precedence restrictions. In this figure the i^{th} node of the precedence diagram represents the **fixing of the component i.** The significance of the numbers shown on the north-east corners of the circles representing tasks will be described later in this chapter.

The task of soldering all the terminals of the components fixed at a work station (refer step 5) comprises of several tasks where each task consists of soldering the terminals of a component. The following relationship is assumed to hold good.

$$S'_{12....n} = S'_1 + S'_2 + ... + S'_n$$

where, $S'_{12....n}$: total soldering time for soldering the terminals of components 1,2,...,n, all done together.

S'_i : soldering time for soldering the terminals of i^{th} component only.

The above relationship suggests that the soldering times are additive in nature. The same type of assumption is made for trimming also (refer step 8) i.e.,

$$T'_{12\ldots n} = T'_1 + T'_2 + \ldots\ldots + T'_n$$

where $T'_{12\ldots n}$ = total trimming time for trimming the terminals of the components 1,2,...,n all done together

T'_i = trimming time for trimming the terminals of i^{th} component.

Table I shows the distribution of the tasks among the six operators as per the practice currently being followed at J.K. Electronics. The tasks which are exactly identical for all the work stations, are listed at the end of the table. These tasks are not included in the precedence diagram. This is because the sum of times for the tasks common to all the work stations can be directly subtracted from the total cycle time of the line in order to find the time available for the allocation of the other tasks to the work stations.

The following notations have been used in the table :

P_{ih} : Bend the terminals of the component by hand before inserting it.

P_{oh} : Bend the terminals of the component by hand after inserting it.

P_{ip} : Bend the terminals of the component by pliers before inserting it.

P_{op} : Bend the terminals of the component by pliers after inserting it.

S_t : Straighten the component by pliers

I_n : Insert the component in the specified hole in the plate.

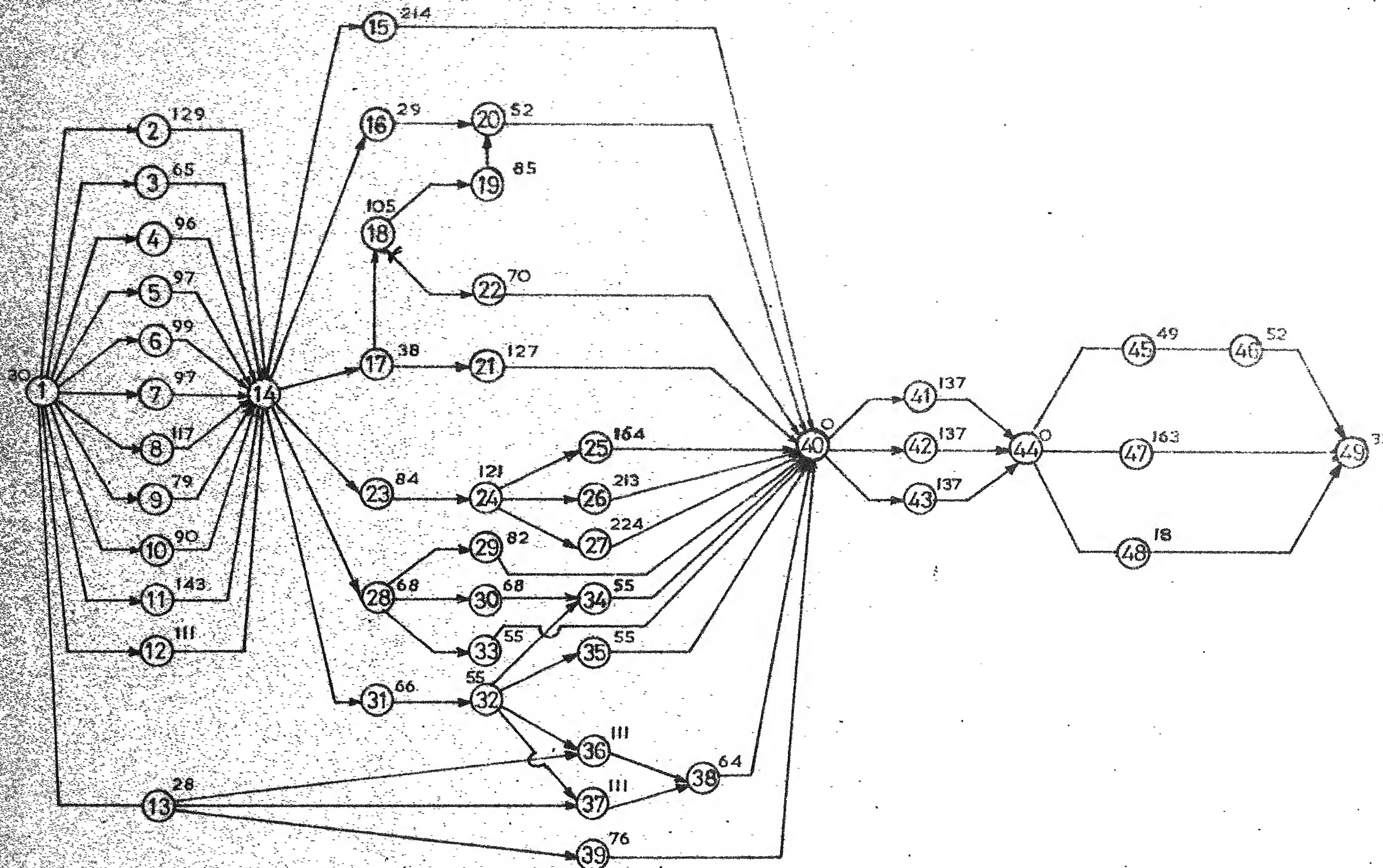


FIG.6 PRECEDENCE DIAGRAM FOR VIF-SIF ASSEMBLY LINE

TABLE I

S.No.	Corresponding component numbers	Task Description	Mean observed task time (in seconds)	Standard Deviation (in seconds)	ψ^2
1	2	3	4	5	6
Station No.1					
1.	1	Write the identification number on the plate	15.45	2.31	7.29
2.	2	Pick up 3.3 K Registers (3)*, P_{ih}, I_n	25.47	3.35	8.13
3.	3	Pick up 4.7 Registers(2), P_{ih}, I_n	13.12	1.97	8.24
4.	4	Pick up 680 Ω Registers(3), P_{ih}, I_n	19.23	2.55	10.77
5.	5	Pick up 1.0 K Registers (3), P_{ih}, I_n	18.59	2.90	3.20
6.	6	Pick up 560 Ω, 2.2K, 56K Registers, P_{ih}, I_n	18.90	3.17	0.34
7.	7	Pick up 10K Registers (3), P_{ih}, I_n	19.85	3.19	1.62
8.	-	Solder** all the terminals (36) of station 1 components	121.51	13.50	0.96
9.	-	Trim*** all the terminals(36), of station 1 components	37.47	4.93	4.85
Station No.2					
10.	11	Pick up 560 Ω (2), 2.2K Registers, P_{ih}, I_n, P_{oh}	31.30	4.19	4.59

(Contd....)

* The figure in the parenthesis indicates the number of similar components which have been combined to form one task.

** This time value does not include the time for picking up soldering rod and placing it back to its place.

*** This time value does not include the time for picking up the pliers and placing it back to its place.

TABLE I (CONTD.)

1	2	3	4	5	6
11.	15	Pick up 0.01 MF Polyster Condensers(3), S_t, P_{ip}, I_n, P_{op}	64.39	6.16	4.44
12.	12	Pick up 1.0M Registers (2), S_t, P_{ip}, I_n, P_{op}	22.18	1.90	10.67
13.	10	Pick up 220 Ω, 680 Ω Registers, S_t, P_{ip}, I_n, P_{op}	18.50	2.30	10.34
14.	8	Pick up 330 Ω (2), 0.2K Registers, P_{ih}, P_{oh}	18.98	1.91	6.18
15.	9	Pick up 680K, 3.9K Registers, P_{ih}, P_{oh}	13.11	1.63	7.53
16.	31	Pick up 25MF Condenser, S_t, P_{ip}, I_n, P_{op}	19.07	2.15	8.71
17.	-	Solder all the terminals (36) of station 2 components	143.82	18.87	2.79
18.	-	Trim all the terminals (36) of station 2 components	45.67	4.50	1.34
Station No.3					
19.	32,33, 34,35	Pick up Pin up Capacitors(2), I_n, P_{oh}	12.19	2.49	6.68
20.	18	Pick up Peaking Coil 1, I_n, drop some molten wax over it	17.92	3.07	0.975
21.	29	Pick up S-Disk type Capacitor(1), I_n, P_{oh}	12.45	2.28	7.03
22.	22	Pick up Peaking coil 2, I_n, P_{oh}	27.51	3.52	0.48
23.	28, 30	Pick up Disk type Capacitors(5), I_n, P_{oh}	27.94	4.76	0.38
24.	-	Solder all the terminals(36) of station 3 components	119.84	15.84	1.99
25	-	Trim all the terminals (36) of station 3 components	52.06	5.21	1.78
26	46	Pick up back Pin up capacitor, I_n, Solder	14.44	3.00	2.38

(Contd....)

TABLE I (CONTD.)

1	2	3	4	5	6
11.	15	Pick up 0.01 MF Polyster Condensers(3), S_t, P_{ip}, I_n, P_{op}	64.39	6.16	4.44
12.	12	Pick up 1.0M Registers (2), S_t, P_{ip}, I_n, P_{op}	22.18	1.90	10.67
13.	10	Pick up 820 Ω, 680 Ω Registers, S_t, P_{ip}, I_n, P_{op}	18.50	2.30	10.34
14.	8	Pick up 330 Ω (2), 0.2K Registers, P_{ih}, P_{oh}	18.98	1.91	6.18
15.	9	Pick up 680K, 3.9K Registers, P_{ih}, P_{oh}	13.11	1.63	7.53
16.	31	Pick up 25MF Condenser, S_t, P_{ip}, I_n, P_{op}	19.07	2.15	8.71
17.	-	Solder all the terminals (36) of station 2 components	143.82	18.87	2.79
18.	-	Trim all the terminals (36) of station 2 components	45.67	4.50	1.34
Station No.3					
19.	32,33, 34,35	Pick up Pin up Capacitors(2), I_n, P_{oh}	12.19	2.49	6.68
20.	18	Pick up Peaking Coil 1, I_n, drop some molten wax over it	17.92	3.07	0.975
21.	29	Pick up S-Disk type Capacitor(1), I_n, P_{oh}	12.45	2.28	7.03
22.	22	Pick up Peaking coil 2, I_n, P_{oh}	27.51	3.52	0.48
23.	28, 30	Pick up Disk type Capacitors(5), I_n, P_{oh}	27.94	4.76	0.38
24.	-	Solder all the terminals(36) of station 3 components	119.84	15.84	1.99
25	-	Trim all the terminals (36) of station 3 components	52.06	5.21	1.78
26	46	Pick up back Pin up capacitor, I_n, Solder	14.44	3.00	2.38

(Contd....)

TABLE I (CONTD.)

1	2	3	4	5	6
Station No.4					
27.	36, 37	Pick up Trap coils(3), S_t, I_n, flip the plate, solder them.	78.06	10.01	1.19
28.	24	Pick up 100KPF Condenser(3), I_n, P_{oh}	18.44	2.61	4.98
29.	23	Pick up 15KPF, 3.3KPF Condensers, I_n, P_{oh}	14.33	2.26	6.08
30.	19, 21	Pick up PF's(5), I_n, P_{oh}	35.95	6.99	1.57
31.	13	Pick up Jumper 2, put wire sleeve, P_{ih}, I_n, P_{oh}	9.06	1.45	4.78
32.	-	Solder the terminals of station 4 components (except No.28)	107.91	15.80	5.75
33.	-	Trim the terminals of station 4 components (except No.28)	29.92	3.11	4.32
34.	38	Pick up Preset Potentiometer, S_t, I_n, flip the plate, solder it, flip the plate again.	30.13	5.45	1.73
35.	17	Pick up Jumper 1, P_{ih}, I_n, P_{oh}	17.51	2.42	3.32
Station No.5					
36.	39	Pick up Zener Diode, P_{ih}, I_n, P_{oh}	13.22	1.81	4.46
37.	16	Pick up OA79Diode, P_{ih}, I_n, P_{oh}	13.20	1.90	3.46
38.	26	Pick up white capped Transistor, P_{ih}, inset it.	37.44	5.68	1.72
39.	25	Pick up black Transistor, P_{ih}, insert it.	15.00	2.33	1.08
40.	27	Pick up Integrated Circuit, bend its terminals by hand at a proper angle, insert it in a particular direction mentioned on it.	75.12	9.12	2.64

(Contd....)

TABLE I (CONTD.)

1	2	3	4	5	6
41.	-	Solder all the terminals (30), of the station 5	91.99	13.44	9.34
42.	-	Trim the terminals of the two Diodes	8.22	0.54	5.64
43.	45	Pick up a 25MF Condenser, S_t, trim its terminals and solder it to the plate	24.06	4.24	6.82
Station No.6					
44.	41, 42, 43	Pick up VIF(2), fix them on the plate	29.89	3.41	8.29
45.	47	Pick up SIF(2), fix them on the plate	42.01	4.12	0.11
46.	20	Pick up the can and fix it on the plate	19.85	2.22	2.62
47.	-	Solder all the terminals of VIF's and SIF's	163.01	16.32	5.76
48.	49	Note down the identification number of the plate	17.95	1.96	1. 53
49.	48	Pick up and solder the back jumper	8.94	1.42	3.94
Tasks common to all the work stations					
50.	-	Pick up a fresh plate from the buffer	7.12	1.42	4.75
51.	-	Pick up the soldering rod, bring it near the plate, then keep it back to its place.	19.90	4.43	3.21
52.	-	Pick up the pliers, bring it near the plate, then keep it back to its place.	12.19	2.49	6.68
53.	-	Flip the plate, flip it again to its original position.	8.57	1.20	9.24

5.1.1. Determination of the time contents for various tasks :

To determine the time contents and their variabilities for the various tasks performed on the VIF-SIF assembly line it is necessary to conduct a time study because this information is not currently available from the factory.

In order to have a statistically sound estimate of the time contents and associated variabilities for the various tasks, it is required that enough time study observations are made. The procedure for the determination of number of observations is discussed in the following section.

5.1.1.1. Statistical Determination of the Number of Readings required for the Time Study -

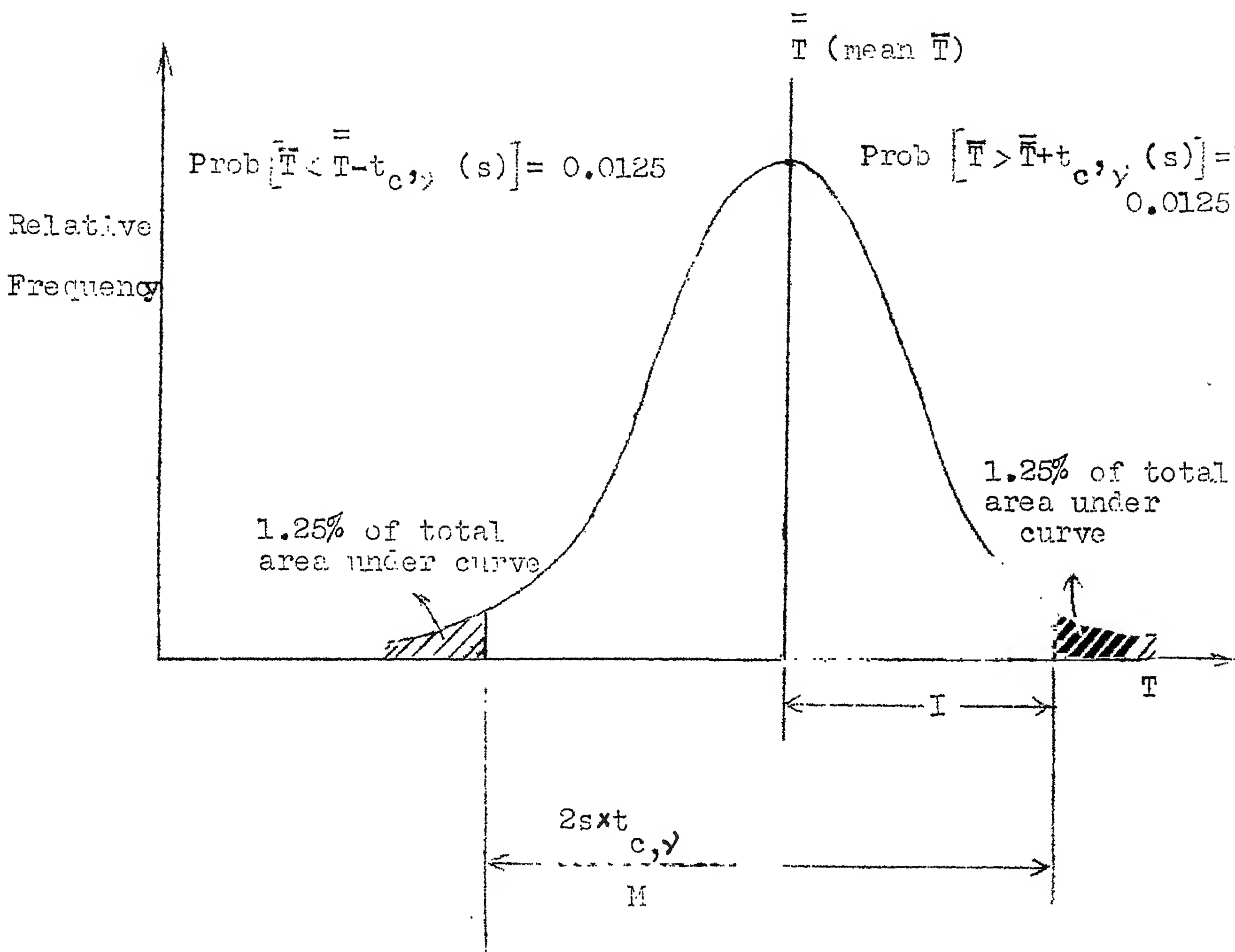
The task times are assumed to follow a normal distribution. This assumption has been validated in the next section. The following procedure has been adopted to determine the number of readings for task time required for each task. The reader may refer the reference [23] for complete details of the procedure.

1. Choose a value for confidence level c, and confidence interval I (refer Fig. 7). The values of c and I have following interpretation :

   $$\text{Prob}\left[\bar{T} - I \leqslant \bar{\bar{T}} \leqslant \bar{T} + I\right] = c$$

   where, $\bar{T}$: sample mean

   $\bar{\bar{T}}$: population mean



T (mean T)
Prob [T < T-t c,γ (s)] = 0.0125
Prob [T > T+t c,γ (s)] =
0.0125
Relative
Frequency
1.25% of total
area under curve
1.25% of total
area under
curve
T
I
2s×t c,γ
M

Fig. 7

2. Take M number of readings for the task.

3. Calculate the sample standard deviation s, as follows:

$$S = \sqrt{\frac{T^2 - \frac{(\Sigma T)^2}{M}}{M-1}}$$

4. Compute the number of readings required **N**, as follows:

$$N = \frac{t^2_{c,\nu} \; s^2}{I^2}$$

where, $t_{c,\nu}$: area under t-distribution curve for confidence level of c and degree of freedom of ν . Here ν is equal to value (M-1).

5. If $N > M$, take N-M more readings for task trim under consideration.

For the time study of the VIF-SIF assembly line following values of parameters chosen are

$c = 0.975$; $I = 0.025$ of $\overline{T}$; $M = 40$; $t_{0.975,39} = 2.04$.

Table I shows the mean and the standard deviation of the task times which are mentioned against the corresponding tasks.

The data for interfailure and repair times of the work stations were also collected. The details of the failure and repair characteristics are given in section 5.2.

5.1.1.2. Chi-Square Test for the Task Times Distribution -

As mentioned previously, the task times are assumed to be following normal distribution. This assumption is validated in this section using ψ^2 test. For the complete details of this test, the reader is referred to reference [6]. The computer listing, in appendix B, includes a general program for ψ^2 test for normally distributed random numbers. The value of ψ^2 is listed in Table I for each of the tasks of the VIF-SIF assembly line. The critical value of ψ^2 at confidence level of 97.5% and a degree of freedom of 6 is equal to 14.4. From Table I, one observes that for all the tasks the values of ψ^2 is less than the critical value of ψ^2. This confirms that all the task times are following normal distribution.

5.2 FAILURE CHARACTERISTICS

5.2.1 Interfailure time-distribution :

Table II shows the first and second moments of interfailure times for each work station. In this table, it is observed that for each work station, the values of the first and the second moments of interfailure times are nearly equal. This gives an indication that these times are exponentially distributed. To confirm Chi,^{2} test has been applied to these variables.

Application of ψ^2 test needs the values of the parameters of interfailure times for each of the station. These parameters have been computed by Regression Analysis for each of the stations. The exact procedure for this estimation of parameter is explained in appendix A.

Table II also shows the parameter of the interfailure time for each of the work station. The corresponding values of ψ^2 are also listed against them.

TABLE II

Work Station No.	X_1*	X_2**	λ	ψ^2
1	621	615	$-\frac{1}{630}$	1.57
2	624	601	$-\frac{1}{624}$	2.96
3	651	633	$-\frac{1}{640}$	10.96
4	639	620	$-\frac{1}{625}$	10.71
5	632	626	$-\frac{1}{634}$	8.67
6	627	618	$-\frac{1}{623}$	9.23

* X_1 is the first moment of the interfailure time of the work station.

** X_2 is the second moment of the interfailure time of the work station. The values are in centiminutes for both X_1 and X_2.

The critical value of ψ^2 at 97.5% confidence level and a degree of freedom of 10 is 20.48. This value is greater than the individual value of ψ^2 for each work station. This further confirms that the interfailure times follow a negative exponential distribution.

5.2.2 Repair Time Distribution :

The repair time is the time taken to repair a failed work station. Broadly speaking, it is the time period after which the failed work station starts functioning again. In the present study, the repair time is the time for which the operator is out of the assembly line for his personal needs. This time is called repair time because after this much period of time the work station resumes functioning as the operator starts working. The time taken to satisfy the personal needs of the operators are nearly similar for each of them because the nature of their personal needs is nearly similar. Therefore it is assumed that the parameters of the repair time distribution for all the work stations are equal.

With this assumption, following are the statistics of the repair time distribution :

1. Mean of the repair time = 11.10 minutes
2. Standard Deviation of the Repair time =2.85 mts.
3. ψ^2 = 5.75
4. Critical value of ψ^2 i.e., ψ^2_{crit} = 14.6 for confidence level of 97.5% and degrees of freedom of 6.

Because ψ^2 is less than ψ^2_{crit}, it is concluded that the repair time follows normal distribution with a mean value of 11.10 minutes and a standard value of 2.85 minutes.

5.3 ALLOWANCES GIVEN TO THE OPERATORS

The following allowances have been given for each task of the assembly line. These values of the allowances are as per the current practice in the factory being considered.

1. Allowance for close attention = 2% of the task time
2. Allowance for mental strain = 1% "
3. Contingency allowance = 4% "

Total allowance = 7% "

After alloting the allowances the standard task times are computed using the following expression :

$$\text{Standard time} = [\text{Mean value of the observed time} \times \text{Rating Factor}] + \text{Allowance}$$

where,

$$\text{Rating factor} = \frac{\text{Observed rate of work(observed output in the shift)}}{\text{Normal rate of work (normal output in the shift)}}$$

5.4 TESTING OF THE METHODOLOGY OF VIF-SIF ASSEMBLY LINE

From the above discussions it is clear that VIF-SIF Assembly Line satisfies the following assumptions :

1. It is a single model assembly line,
2. Only one operator is assigned to each work station,
3. The task times, and hence the work station times, follow normal distribution,
4. The interfailure times follow negative exponential distribution, and
5. The repair time follow the normal distribution.

Thus the assembly line satisfies the assumptions made during the development of the methodology. Therefore, the methodology can be applied to this assembly line.

In Phase I, while carrying out the balancing of the assembly line it should be borne in mind that the soldering and trimming operations associated with the components assigned to a work station have to be necessarily completed in the work station before the plate is passed onto the next station. This means that whenever a task of fixing a component is assigned to an operator, the task times for soldering and trimming, alongwith the task time for fixing it, are subtracted from the time available with with the operator. To facilitate the understanding of the balancing procedure, a new term called 'component time' has been introduced. Component time for component i is equal to the summation of component's fixing time,

soldering time and the trimming time. Mathematically, it can be expressed by the following relationship :

$$c'_i = c'_{fi} + s'_i + T'_i$$

where, c'_{fi} : Task time for fixing the i^{th} component

s'_i : Soldering time of the i^{th} component. This time does not include the time for bringing the soldering rod to the plate and the time for putting it back to its place.

T'_i : Trimming time of the i^{th} component. This time does not include the time for bringing the pliers to the plate and the time for putting it back to its plate.

The component times for the various components are shown in the precedence diagram on the north-east corner of the circles representing the various tasks. The methodology has been tested on the VIF-SIF assembly line. The results obtained are presented in the next chapter.

CHAPTER VI

RESULTS AND DISCUSSIONS

The results obtained by using the methodology on the data of 'VIF-SIF' Assembly Line of J.K.Electronics, Kanpur are presented in the following sections. In conformity with the generalized formulation of the problem given in chapter 3, the results are discussed separately for both the phases.

6.1 PHASE I

Table III depicts the results obtained for the allocation of the tasks to the various work stations by using the methodology for this phase. The work loads mentioned against the work stations do not include the task time for the tasks which are identical for all the work stations. An examination of the work load values given in the table indicates that the work load is distributed quite uniformly among the various stations. Further, it yields a Balance Delay Ratio of 1.20%, a Smoothness Index of 13.64 and a cycle time of 725 centiminutes.

These results are compared with the results of the current practice for loading the work stations. Table III also shows the current allocations of the tasks to the various work stations. It is observed that the

[illegible]

[illegible] OF THE TASKS TO [illegible] VARIOUS [illegible]

WORK STATION NO.	THE LIST OF TASKS ASSIGNED TO THE WORK STATION	[illegible]
[illegible] THE [illegible] METHODOLOGY		
[illegible]	[illegible]-9-[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]-7-[illegible]-9	[illegible]
[illegible]	[illegible]-7-[illegible]	[illegible]
[illegible]	[illegible]-7-[illegible]	[illegible]
[illegible]	[illegible]-44-47-[illegible]-48-[illegible]9	[illegible]
		[illegible]
		S.I.= [illegible]
THE CURRENT PRACTICE		
[illegible]	[illegible]-5-6-7	[illegible]
[illegible]	[illegible]-9-[illegible]-3[illegible]	[illegible]9
[illegible]	[illegible]-28-29-3[illegible]-[illegible]-35-36	[illegible]
4	[illegible]-24-[illegible]-27-2[illegible]	[illegible]9.
5	[illegible]-[illegible]5-[illegible]6-[illegible]7-[illegible]9-[illegible]5	7[illegible].
6	4[illegible]-4[illegible]-43-4[illegible]7-4[illegible]-49	[illegible]
		B.D.K.=[illegible] P[illegible]
		S.I.=[illegible]94.[illegible]

line is not balanced for these allocations because station 2 is much more heavily loaded as compared to the other work stations and therefore is a bottleneck for the line. From the results one notes that the assembly line has a Balance Delay Ratio of 11.30%, a Smoothness Index of 594.1 and the cycle time of 809 centiminutes.

Thus it is observed that the use of the methodology reduces the Balance Delay Ratio from 11.30% to 1.20%, indicating approximately 10.1% improvement in the efficiencies of the work stations. The cycle time reduces from 809 centiminutes to 725 centiminutes, indicating a 11.5% reduction. The small value of Smoothness Index indicates that the total idle time of the line is distributed much more uniformly compared to the one in the current practice.

The other alternative solutions representing equally good balances are shown in Table IV. The Smoothness Index values for these solutions range from 13.78 to 15.56, for a Balance Delay Ratio of 1.20%.

6.2 PHASE II

The output of Phase I is the input to Phase II. In Phase I, the tasks are allocated to the various work stations while in phase II, the optimum buffer capacities for the buffer spaces are determined.

TABLE IV

ALTERNATIVE SOLUTIONS

WORK STATION NO.	T[illegible] OF TASKS [illegible] TO TH[illegible] WORK STATION	TOTAL WORK L[illegible] TH[illegible] W[illegible] STATION ([illegible])
SOLUTION NO. [illegible]	S.I.= [illegible]	
[illegible]	[illegible]-9	[illegible]
[illegible]	2-[illegible]-7-[illegible]-4-[illegible]-6	[illegible]
[illegible]	[illegible]-7-[illegible]-2	[illegible]
[illegible]	[illegible]-2-7	[illegible]
[illegible]	27-[illegible]-8-[illegible]-4	[illegible]
[illegible]	4[illegible]-47-[illegible]-48-49	[illegible]
SOLUTION NO. [illegible]	S.I.= [illegible]	
[illegible]	1-[illegible]-9-[illegible]-13	7[illegible]
[illegible]	3-[illegible]-5-[illegible]-29-[illegible]-4-[illegible]-31-[illegible]7	[illegible]
[illegible]	3[illegible]-[illegible]-34-[illegible]	7[illegible]
4	2[illegible]-3[illegible]-5-[illegible]44	7[illegible]
5	27-[illegible]9-[illegible]9-2[illegible]-37-[illegible]-38-4	[illegible]
6	4[illegible]-47-45-[illegible]-48-49	7[illegible]
SOLUTION NO. [illegible]	S.I.= [illegible]	
[illegible]	[illegible]-5-[illegible]	[illegible]
2	7-[illegible]-9-[illegible]9-[illegible]-16-[illegible]	[illegible]
3	2[illegible]-7-[illegible]2-[illegible]	[illegible]
[illegible]	[illegible]-[illegible]7-[illegible]	[illegible]
[illegible]	2[illegible]-3[illegible]-9-[illegible]9-[illegible]-27-[illegible]-33-[illegible]	7[illegible]
6	4[illegible]-45-[illegible]-49	[illegible]

TABLE IV(CONTD.)

ALTERNATIVE SOLUTIONS

WORK STATION NO.	THE LIST OF TASKS ASSIGNED TO THE WORK STATION	TOTAL WORK LOAD OF THE WORK STATION (IN DECI-MINUTES)
SOLUTION NO. 4	S.I.=[illegible]	
[illegible]	[illegible]	[illegible]
[illegible]	6-[illegible]	[illegible]
[illegible]	[illegible]-6-[illegible]-35	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]-9-[illegible]6-[illegible]-4	[illegible]
6	4[illegible]-[illegible]-7-[illegible]5-[illegible]6-[illegible]-49	[illegible]
SOLUTION NO. 5	S.I.=[illegible]	
1	2-[illegible]-7-[illegible]-9-4-[illegible]	[illegible]
2	3-5-[illegible]-9-[illegible]9-[illegible]-20-21-17	7[illegible]
3	29-[illegible]-32-37-[illegible]8-33-[illegible]	7[illegible]
4	[illegible]5-[illegible]9-[illegible]-22-38	7[illegible]
5	24-[illegible]-40	7[illegible]
6	42-[illegible]-47-45-46-48-49	7[illegible]5
SOLUTION NO. 6	S.I.=[illegible].36	
[illegible]	[illegible]-5	[illegible]
[illegible]	4-5-[illegible]-9-[illegible]-16-[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
4	27-[illegible]	[illegible]
5	3[illegible]-[illegible]9-[illegible]-36-3[illegible]	[illegible]9
6	4[illegible]-49	[illegible]

TABLE IV (CONTD.)

ALTERNATIVE SOLUTIONS

WORK STATION NO.	THE LIST OF TASKS ASSIGNED TO THE WORK STATION	TOTAL WORK LOAD OF THE WORK STATION (IN CENTI-MINUTES)
SOLUTION NO. 7 S.I.= [illegible]		
[illegible]	1-7-[illegible]	7[illegible]
[illegible]	[illegible]-5-6-[illegible]-5	[illegible]
[illegible]	36-[illegible]-7-37	7[illegible]
4	[illegible]	7[illegible]
5	29-[illegible]	7[illegible]
6	[illegible]-7-[illegible]-49	7[illegible]
SOLUTION NO. [illegible] S.I.= [illegible]		
[illegible]	1-[illegible]	[illegible]
[illegible]	4-7-[illegible]	7[illegible]
[illegible]	3-[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
5	[illegible]-7-[illegible]	[illegible]
6	4[illegible]-5-7-[illegible]-9	7[illegible]
SOLUTION NO. 9 S.I.= [illegible]		
[illegible]	1-2-[illegible]-9-[illegible]	7[illegible]
[illegible]	5-6-[illegible]-17	7[illegible]
[illegible]	2[illegible]-5	7[illegible]
4	3[illegible]-15-3[illegible]	7[illegible]
5	27-25-[illegible]-26-[illegible]	7[illegible]
6	4[illegible]-4[illegible]-4[illegible]-44-47-45-[illegible]-46-49	7[illegible]

Table V shows the mean and the standard deviation of the work station times. The mean of the work station times shown in the table also include the task times for the tasks which are exactly identical for all the work stations (e.g., the task of bringing the soldering rod near the plate and putting it back to its position, etc.).

Table VI shows the values of the number V_i associated with i^{th} buffer space. It may be recalled that V_i is the summation of variances of the i^{th} and $(i+1)^{th}$ work stations. Three groups of buffer spaces have been obtained using the procedure explained in section 4.2.2. Group 1 consists of buffer space 1, group 2 consists of buffer spaces 4 and 2, and group 3 consists of buffer spaces 3 and 5. The three groups of buffer capacities yield three design variables.

Table VII shows the results obtained after applying the methodology for phase II. All the design points reached during the search for optimal solution have been tabulated there. The starting point for the optimization procedure used is (0,0,0) for this table. The components of the design point (x_1, x_2, x_3) are the values of the buffer capacity sizes associated with the group 1,2 and 3 of buffer spaces, respectively. For the results presented in table VII to table XI the confidence level chosen is 97.5%. Table VIII represents the results obtained when the methodology was applied using

TABLE V

Work Station No.	Mean Station Time (in centiminutes)	Corresponding Standard Deviation (in centiminute)
1	816	105
2	821	106
3	812	131
4	822	114
5	813	119
6	827	138

TABLE VI

V_i (arranged in increasing order)	226.1	271.6	275.2	301.6	332.0
Corresponding Buffer space number	1	4	2	3	5

TABLE VII

S. NO.	DESIGN POINT	OUTPUT RATE PER SHIFT	TOTAL COST PER UNIT	PERCENT UTILIZATION OF THE WORK STATION					
				1	2	3	4	5	6
1	0,0,0	18.26	88.32	81.3	69.2	61.8	55.4	44.2	41.2
2	2,0,0	30.08	86.83	82.5	72.8	64.9	57.1	48.9	42.5
3	6,0,0	32.12	85.82	81.7	73.2	66.3	59.8	52.4	44.3
4	14,0,0	32.43	85.43	84.4	73.5	70.4	61.2	53.6	45.2
5	30,0,0	32.98	85.92	83.2	83.6	71.2	62.[illegible]	54.7	46.2
6	14,2,0	33.02	85.32	84.3	74.2	71.3	61.7	54.2	46.1
7	14,6,0	33.87	85.03	84.7	75.9	72.4	63.8	58.2	49.1
8	14,14,0	34.61	84.73	84.2	76.7	73.7	64.3	58.1	50.3
9	14,30,0	35.89	84.77	85.8	83.2	76.2	68.3	69.1	52.1
10	14,14,2	35.04	84.46	83.7	77.2	72.3	65.1	59.3	53.6
11	14,14,6	36.10	87.82	83.0	78.3	73.4	66.3	61.3	55.4
12	14,14,14	37.71	82.93	83.8	79.7	76.5	68.5	63.2	58.0
13	16,14,6	36.24	82.55	83.7	79.9	73.7	67.3	61.5	56.4
14	20,14,6	37.21	82.95	83.8	80.3	74.3	67.7	61.9	56.8
15	16,16,6	37.13	82.73	83.5	79.3	74.1	67.4	61.3	56.5
16	16,14,8	38.07	81.98	84.3	80.2	75.6	68.2	63.4	57.3
17	16,14,12	38.43	81.23	85.1	80.4	75.9	69.2	64.1	58.2
18	16,14,20	40.39	81.53	85.7	81.8	77.2	72.1	66.3	61.2
19	18,14,12	39.13	80.75	85.4	80.9	76.7	71.2	65.2	60.0
20	22,14,12	39.14	81.12	85.5	81.2	76.9	72.0	65.6	60.3
21	18,16,12	39.17	81.23	85.5	81.3	77.3	71.7	65.4	60.4
22	18,14,14	39.19	81.43	85.7	81.2	77.5	71.9	65.7	60.8
23	[illegible],14,14	39.27	80.93	85.8	81.4	78.[illegible]	70.[illegible]	[illegible]	62.5

TABLE VIII

S. NO.	DESIGN POINT	OUTPUT RATE PER SHIFT	TOTAL COST PER UNIT	PERCENT UTILIZATION OF THE WORK STATIONS					
				1	2	3	4	5	6
1	0,0,0	28.70	88.33	82.5	71.9	63.4	55.3	47.3	41.6
2	2,0,0	3[illegible].54	86.65	83.1	73.9	65.8	57.[illegible]	49.2	42.7
3	6,0,0	32.41	85.68	82.4	74.3	68.9	61.6	54.4	44.9
4	14,0,0	32.87	85.35	84.8	74.5	71.9	61.9	55.7	46.8
5	30,0,0	33.41	85.74	84.7	84.2	73.4	63.[illegible]	56.6	47.7
6	14,2,0	33.13	85.21	84.8	75.9	72.7	62.6	56.3	47.4
7	14,6,0	34.33	84.93	85.1	77.1	72.8	65.2	59.2	50.7
8	14,14,0	35.01	84.62	84.6	78.0	73.3	67.1	60.3	52.3
9	14,30,0	36.13	84.86	85.4	85.1	76.3	69.3	65.2	53.3
10	14,14,2	35.28	84.29	84.5	79.3	73.8	67.4	60.9	54.1
11	14,14,6	36.14	82.64	84.2	86.1	74.2	67.6	61.8	56.3
12	14,14,14	38.13	82.87	84.1	80.2	75.3	69.3	63.5	58.1
13	16,14,6	36.62	82.32	84.3	80.7	74.8	68.1	62.1	57.2
14	20,14,6	37.66	82.78	84.6	81.1	75.3	68.2	62.3	57.4
15	16,16,6	37.96	82.42	84.4	80.8	75.9	68.6	62.4	58.6
16	16,14,8	38.48	81.82	85.1	80.9	76.5	69.3	64.6	59.2
17	16,14,12	39.06	81.00	86.9	80.4	76.7	71.9	65.3	60.6
18	16,14,20	41.23	81.22	85.8	82.3	78.2	73.4	67.1	63.3
19	18,14,12	39.41	80.69	86.1	81.2	77.7	72.8	66.6	62.1
20	22,14,12	39.74	81.04	86.2	81.4	77.4	72.8	66.6	62.1
21	18,16,12	39.58	80.97	86.1	81.3	77.4	72.5	66.4	61.4
22	18,14,14	39.62	81.12	86.1	81.3	77.5	72.5	66.4	61.5
23	20,14,14	39.88	80.86	86.3	81.4	77.7	72.8	66.7	61.9

antithetic set of random numbers. Further, the results obtained by taking starting design point as (24,24,24) are tabulated in table IX.

Table X shows the results of the simulation runs when the equal buffer capacities were considered for all the buffer spaces. The second column of the table represents the value of the buffer capacities. It is to be noted in the table that there are two rows for every value of the buffer capacities. The second row represents the results of simulation runs using antithetic set of random numbers. The average results of the two rows for each value of the buffer capacity in Table X are tabulated in Table XI.

6.2.1 Discussions of the Tables :

From Table XI, it is observed that the introduction of buffer inventories considerably affects the rate as well as the cost of production from the assembly line. An examination of the table indicates that the higher the allowed buffer capacity the greater is the rate of production from the assembly line. An increase of buffer capacity from zero to a value 16 for all the work stations (i.e.,to a total buffer capacity value of 16 x 5 = 80 for whole of the line) increases the average rate of production from 28.48 units per shift to 39.07 units per shift, indicating an increment of 26.5%. Further, it is observed that there is less improvement in the production rate after the buffer capacities for

TABLE IX

S. NO.	DESIGN POINT	OUTPUT RATE PER SHIFT	TOTAL COST PER UNIT	PERCENT UTILIZATION OF THE WORK STATIONS 1	2	3	4	5	6
1	24,24,24	42.13	85.98	85.5	84.5	84.1	82.7	75.2	73.0
2	22,24,24	42.[illegible]1	85.62	85.1	84.2	83.7	81.9	74.8	72.8
3	18,24,24	41.79	85.37	84.7	83.6	83.2	80.6	73.5	70.2
4	10,24,24	41.42	85.64	83.2	82.1	82.3	78.3	7[illegible].6	68.4
5	18,22,24	41.58	85.14	83.7	82.8	82.7	78.7	72.1	69.5
6	18,18,24	41.71	84.73	82.4	81.2	80.2	76.4	68.5	67.3
7	18,10,24	40.68	85.07	83.7	80.1	82.3	79.2	73.3	70.6
8	18,18,22	41.23	85.25	84.9	83.0	84.3	77.4	72.5	69.3
9	18,18,18	40.71	83.50	84.7	82.8	82.9	70.9	71.0	67.7
10	18,18,10	40.12	83.81	83.6	82.5	82.1	76.2	70.3	66.2
11	16,18,18	40.43	83.89	83.9	83.2	81.7	76.2	70.7	66.8
12	18,16,18	40.49	82.73	83.8	82.3	81.8	75.2	69.2	66.6
13	18,12,18	39.87	82.94	84.5	82.4	80.2	74.3	68.5	65.2
14	18,16,16	39.71	81.36	86.4	82.3	78.5	73.6	67.2	64.7
15	18,16,12	39.56	80.96	86.0	81.2	77.4	72.6	66.5	61.5
16	18,16,04	37.93	82.36	84.3	78.4	74.2	68.2	61.4	58.3
17	16,16,12	39.49	81.42	85.3	80.4	76.5	69.9	64.8	60.2
18	18,14,12	39.41	8[illegible].69	86.1	81.2	77.7	72.8	66.6	62.1
19	18,10,12	[illegible]8.75	81.23	85.1	79.2	75.2	70.4	67.3	60.2
20	18,14,10	39.22	8[illegible].89	85.1	80.5	76.3	71.7	66.3	61.3
21	16,14,12	39.06	81.00	86.9	80.4	76.7	71.2	65.3	60.6
22	18,12,12	39.12	8[illegible].21	86.9	80.2	76.2	71.3	65.8	61.1

TABLE X

S. NO.	DESIGN VARIABLE	OUTPUT RATE PER SHIFT	TOTAL PER UNIT	PERCENT UTILIZATION OF WORK STATIONS					
				1	2	3	4	5	6
1		28.7	88.08	82.5	71.9	63.4	55.8	47.3	41.6
		28.[illegible]6	88.32	81.3	69.2	61.8	55.4	44.2	41.[illegible]
2	1	30.32	86.75	82.7	73.5	65.4	57.2	48.7	43.2
		29.45	87.08	81.4	69.9	63.8	57.1	49.2	42.7
3	2	31.38	85.77	82.3	75.7	68.3	61.7	53.9	46.0
		30.72	85.92	81.7	73.6	66.4	60.0	52.3	44.7
4	4	32.33	85.44	82.6	76.9	69.4	62.6	54.6	47.3
		32.13	85.71	82.4	75.8	69.2	61.9	54.8	46.3
5	6	33.42	85.32	84.5	77.3	69.4	61.2	55.2	47.7
		33.27	85.46	83.9	75.9	71.5	62.3	55.1	47.2
6	8	34.28	84.53	83.4	77.8	69.4	63.3	57.3	51.[illegible]
		33.78	84.67	84.6	76.2	71.7	62.6	55.6	49.8
7	9	35.14	84.16	84.3	78.5	72.9	66.1	58.7	51.4
		34.67	84.18	84.2	76.8	72.3	64.6	56.3	50.5
8	11	36.79	83.19	83.1	79.9	73.8	66.2	60.6	55.9
		35.86	83.32	83.4	78.1	73.7	65.6	58.4	52.7
9	13	37.65	82.96	84.1	80.7	74.5	68.2	61.6	56.3
		36.96	83.06	83.8	79.1	73.9	66.4	62.5	57.4
10	14	38.13	82.87	84.1	81.2	75.2	69.3	62.5	58.0
		37.71	82.93	83.8	79.7	76.5	68.5	64.2	58.0
11	15	38.76	82.55	84.5	81.7	76.2	70.1	66.3	60.3
		38.42	82.63	83.9	80.9	77.6	70.6	64.7	59.4
12	16	39.31	81.83	83.4	82.1	78.2	72.6	67.2	64.7
		38.82	81.92	83.8	80.9	77.6	72.6	67.7	62.8
13	17	40.21	83.00	84.5	82.3	81.9	75.3	69.8	66.3
		39.79	83.09	84.3	80.8	79.3	75.2	66.4	63.2
14	19	41.17	84.02	84.8	83.3	83.9	78.5	72.3	70.0
		40.93	84.32	83.6	82.6	80.9	78.2	75.3	68.7
15	21	41.86	84.52	85.1	84.1	84.1	82.5	74.4	[illegible]
		41.36	84.87	83.9	82.9	79.7	78.6	75.1	[illegible]
16	24	42.12	85.96	85.5	84.5	84.1	82.7	[illegible]	[illegible]
		41.79	86.12	84.2	83.6	79.9	78.9	[illegible]	[illegible]
17	34	45.24	91.66	89.5	89.1	88.7	86.3	[illegible]	[illegible]
		43.93	92.12	88.7	88.1	86.5	82.6	[illegible]	[illegible]

TABLE XI

S. NO.	DESIGN VARIABLE	OUTPUT RATE PER SHIFT	TOTAL PER UNIT	PERCENT UTILIZATION OF WORK STATIONS					
				1	2	3	4	5	6
1		28.48	88.20	81.9	70.5	62.6	55.6	45.8	[illegible]
2	1	29.88	86.91	[illegible]	71.7	64.6	57.2	48.9	[illegible]
3	2	31.08	85.84	[illegible]	74.6	67.3	60.8	53.1	[illegible]
4	4	32.23	85.58	82.5	76.4	69.3	62.4	54.7	46.8
5	6	33.34	85.39	84.2	76.6	70.4	61.8	55.2	47.4
6	8	[illegible]	84.60	84.0	77.0	70.5	[illegible]	[illegible]	[illegible]
7	9	34.90	84.17	84.2	78.1	72.6	65.3	57.5	50.9
8	11	36.32	83.25	83.2	79.0	73.7	65.9	59.5	54.3
9	13	37.30	83.01	83.9	79.9	74.2	67.3	62.0	56.8
10	14	37.92	82.90	83.9	80.4	75.8	68.8	63.8	58.0
11	15	38.59	82.59	84.2	81.3	76.9	70.4	65.5	59.9
12	16	39.07	81.87	83.6	81.5	77.9	72.6	67.4	63.4
13	17	40.00	83.04	84.4	81.5	80.6	75.2	68.1	64.3
14	19	41.05	84.17	84.2	82.9	82.4	78.3	73.8	69.3
15	21	41.61	84.69	84.5	83.5	81.9	80.6	74.9	70.5
16	24	41.96	86.08	84.8	84.0	82.0	80.8	75.2	72.6
17	34	44.58	91.89	89.1	88.6	87.6	84.4	83.8	82.4

all the work stations have reached a value of 17.

An examination of Table XI indicates that there is an optimum value of the buffer capacities which gives the minimum cost of production per unit. The cost of production per unit reduces from Rs. 88.20 to Rs.81.87 if the buffer capacities for all the buffer spaces are increased from zero to a value sixteen. The above discussion implies that by increasing the buffer capacities for all the buffer spaces from zero to a value 16, an increment of 26.5% in the production rate and a reduction of 7.1% in the cost of production is achieved.

From the same table it is further observed that the first work station is least blocked, for there is an unlimited amount of units available for processing. The last work station is the maximum blocked and shows minimum utilization. This is expected because an occurrence of blocking at any work station increases the probability that it will also occur in the successiding work stations. Because of this cumulative effect, the last work station experiences the maximum blocking.

Further it should be noted that the higher the buffer capacities, the greater is the utilization of the work stations. This is because of the fact that higher buffer capacity reduces the chances of blocking and this in turn yields greater utilization of the work stations.

Table VII shows the optimum solution obtained by using the methodology. The optimum values of

the buffer capacities are

$$B_1=18,\ B_2=14,\ B_3=12,\ B_4=14 \text{ and } B_5=12$$

Corresponding to the optimum values for the buffer capacities, the rate of production is 39.27 units per shift and the cost of production is Rs.80.72 per unit. These values of the production rate and the cost of production are the average values of the results obtained by using the antithetic sets of random numbers tabulated in Table VII and VIII respectively.

The simulation run of the unbalanced assembly line (i.e., the line before balancing in phase I) resulted in the following values. For the same buffer capacity sizes of 35 for all the buffer spaces, the rate of production is 32.42 units per shift and the cost of production per unit is Rs.92.36.

This shows that the methodology developed in this research work helps in reducing the buffer capacities considerably. It also results in an improvement of the rate of production by 17.2% and a reduction in the cost of production by 12.5%.

It was observed from Table XI that if same values of the buffer capacities are used for all the buffer spaces, the optimum result gives a production rate of 39.07 units per shift and cost of production Rs 81.87 per unit. This

shows that if different values are used for the buffer capacities (as is the case in the methodology) instead of using the same values for all the buffer spaces, a reduction of 1.5% in the cost of production and an improvement of 0.5% in the production rate is achieved. The reason for this reduction of cost is attributed to the high buffer holding cost per unit in the last work station.

The efficiency of the methodology can be improved by a proper choice of the initial conditions for the buffer capacity levels. The author has experienced that the initial values of the buffer sizes for a simulation run greatly affect the total computer time required to reach the steady state of the simulation run.

CHAPTER VII

CONCLUSIONS AND SCOPE FOR FURTHER RESEARCH

7.1 CONCLUSIONS

The following inferences are drawn from this study :

1. The design methodology has been divided into two phases. The first phase allocates the tasks to various work stations so as to minimize the Balance Delay Ratio and Smoothness Index for the line. The second phase determines the optimum size of the buffer capacities which minimizes the total cost of production per unit.
2. In phase I, reallocation of tasks to the various work stations improves the overall efficiency of the line significantly. This phase gives more than one solutions which are equally good. Thus several alternative solutions are available for the line supervisor and he can select the one best suited to him from practical considerations.
3. In phase II, introduction of buffers significantly affects the rate of production from the line. Following observations are made in this respect :
   i) The production rate rapidly increases with the increase of the buffer sizes till the buffer size is 17. After this, the improvement in

production rate due to increase in buffer sizes is very meagre.

ii) The blocking decreases as the buffer size decreases.

4. In phase II the results of the study indicate that the introduction of buffers affects the per unit cost of assembly. With the optimum value of buffer obtained from the solution, the rate of production, shows an increment of 17.2% as compared to the present rate of production from the line. The improvement in the performance of the existing system is reflected by a reduction of the cost of production to the extent of 12.5%.

7.2 SCOPE FOR FURTHER RESEARCH

There is ample scope for further research in the field of design of an assembly line system. Attempts may be made to relax some of the assumptions made in the development of the methodology in the present work. A more realistic design of the assembly line would be achieved if the following aspects of the assembly line are also considered :

1. In the methodology, an attempt was made to distribute the total work load of the assembly line equally among the operators, assuming the performance ratings of all the

operators to be same. However, in real life practice, the performance ratings of the various operators are hardly found to be the same. This feature should be incorporated in the methodology if the management is ready to pay the operators according to their individual performance ratings. This will allow the allocation of more work load to the operators having higher performance ratings. Some research work has been reported in this area by Mansoor [26] but it needs further investigations.

2. The modern trends in customers' desire for variety of models and the highly competitive market make the management think in terms of mixed model assembly lines. A generalized formulation of assembly line balancing problem should incorporate this feature. Some research work [37] has been reported in literature which deals with mixed model assembly line but it is very little both in volume and substance.

3. In the proposed methodology it was assumed that all the operators are well trained and their performance ratings do not change with time. It has been observed that in practice 1) new operators are occasionally put on the

line and, 2) old operators are transferred to other departments. Therefore, the concept of 'learning' plays an important role in deciding the overall efficiency of the assembly line. Further research work is needed to study the effect of 'learning process' on the behaviour of the assembly line.

4. It might be interesting to carry out a sensitivity analysis to determine the dependency of the production cost and the rate of production upon the buffer holding costs and the variances of the work stations.

Each of the above mentioned avenues requires further research. Incorporation of these features will help make the design of assembly line system more realistic.

REFERENCES

1. Arcus, Albert L., "COMSOAL : A Computer Method of sequencing Operations for Assembly Lines", from Buffa, E.S., Readings in Production and Operations Management, John Wiley and Sons, Inc., Sydney, 1966.

2. Anderson, David R., and Moodie, C.L., "Transient and Steady State Minimum Cost in-process inventory capacities for Production Lines", Ph.D. Dissertation, School of Industrial Engineering, Purdue University, 1968.

3. Barten, K.A., "A Queuing Simulator for determining Optimum inventory levels in sequential process", Journal of Industrial Engineering, Vol. 13, No. 4, July-Aug., 1962, p 245.

4. Bryton, Benjamin, "Assembly Line Balancing," M.S. Thesis, North-Western University, Evanston, Illinois, June, 1954.

5. Bowman, E.H., "Assembly Line Balancing by Linear Programming," Operations Research, Vol. 8, No.3, May-June, 1960, p 385.

6. Burford, Roger L., Statistics : A Computer Approach, Charles E. Merrill Publishing Company, Columbus, Ohio, 1968.

7. Das, S.K., and Sahu, K.C., "A Comprehensive Heuristic Approach to Assembly Line Design," Industrial Engineering and Management, Vol. 6, No. 3 and 4, Jan.-June, 1971, p 41.

8. Demyanyuk, F.S., The Technological Principles of Flow Line and Automated Production, Vol. I, The Macmillan Co., New York, 1963.

9. Fox, Richard L., Optimization Methods for Engineering Design, Addison-Wesley Publishing Co., Massachusetts, 1973.

10. Freeman, D.R., and Jucker, J.V., "The Assembly Line Balancing Problem, "Journal of Industrial Engineering, Vol. 18, No. 6, June, 1967, p 361.

11. Freeman, Michael C., "The Effects of Breakdowns and Inter-stage storage on Production Line Capacity," Journal of Industrial Engineering, Vol. 15, No.4, July-Aug., 1964, p 194.

12. Gutjahr, A.L., and Nemhauser, G.L., "An Algorithm for the Line Balancing Problem, "Management Science, Vol. 11, No. 2, Nov., 1964, p 308.

13. Held, M., Karp, R.M., and Shareshian, R., "Assembly Line Balancing - Dynamic Programming with Precedence Constraints," Operations Research, Vol. 11, No.3, May-June, 1963, p 442.

14. Helgeson, W.B., and Birnie, D.P., "Assembly Line Balancing using the Ranked Positional Weight Technique," Journal of Industrial Engineering, Vol. 12, No. 6, Nov.-Dec., 1961, p 394.

15. Hoffman, T.R., "Assembly Line Balancing with a Precedence Matrix," Management Science, Vol. 9, No. 4, July, 1963, p 551.

16. Hu, T.C., "Parallel Sequencing and Assembly Line Problems," Operations Research, Vol. 9, No. 6, Nov.-Dec., 1961, p 341.

17. Ignall, E.J., "A Review of Assembly Line Balancing," Journal of Industrial Engineering, Vol. 16, No.4, July-Aug., 1965, p 244.

18. Jackson, J.R., "A Computing Procedure for a Line Balancing Problem," Management Science, Vol. 2, No.3, April, 1956, p 261.

19. Kilbridge, M.D., and Wester, L., "A Henristic Method of Assembly Line Balancing," Journal of Industrial Engineering, Vol. 12, No. 4, July-Aug., 1961, p 292.

20. Kilbridge, M.D., and Wester, L., "Heuristic Line Balancing - A Case," Journal of Industrial Engineering, Vol.13, No.3, May-June, 1962, p 139.

21. Kilbridge, M.D., and Wester, L., "A Review of Analytical Systems of Line Balancing," Operations Research, Vol. 10, No. 5, 1962, p 626.

22. Klein, M., "On Assembly Line Balancing," Operations Research, Vol. 11, No. 2, March-April, 1963, p 274.

23. Krick, Edward V., Methods Engineering, John Wiley & Sons, Inc., New York, London, 1962.

24. Mansoor, E.M., "Assembly Line Balancing - An Improvement on the Ranked Positional Weight Technique," Journal of Industrial Engineering, Vol. 15, No.2, March-April, 1964, p 73.

25. Mansoor, E.M., and Ben-Tuvia, S., "Optimizing Balanced Assembly Lines," Journal of Industrial Engineering, Vol. 17, No.3, March, 1966, p 126.

26. Mansoor, E.M., "Assembly Line Balancing-A Heuristic Algorithm for Variable Operator Performance Levels," Journal of Industrial Engineering, Vol. 19, No.12, Dec., 1968, pp 618.

27. Master, Anthony A., "An Experimental Investigation and Comparative Evaluation of Production Line Balancing Techniques," Management Science, Vol. 16, No.7, July, 1970, p 728.

28. Moodie, C.L., and Young, H.H., "A Heuristic Method for Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times", Journal of Industrial Engineering, Vol. 16, No.1, Jan.-Feb., 1965, p 23.

29. Naylor, Thomas H., Computer Simulation Techniques, John Wiley & Sons, Inc., New York, 1966.

30. Naylor, Thomas H., et al, Computer Simulation Experiments with Models of Economic Systems, John Wiley & Sons, Inc., New York, 1971.

31. McMillan, Clande and Gonzalez, Richard F., Systems Analysis : a computer approach to decision models, Richard D. Irwin, Inc., Homewood, Illinois, Revised Edition, 1968.

32. Ramsing, K., and Downing, R., "Assembly Line Balancing with Variable Elemental Times," <u>Industrial Engineering</u>, No. 1, Jan., 1970, p 41.

33. Salveson, M.E., "The Assembly Line Balancing Problem," <u>Journal of Industrial Engineering</u>, Vol. 6, No.3 May-June, 1955, p 18.

34. Smith John, M.A., Computer Simulation Models, Hafner Publishing Co., New York, 1968.

35. Tonge, F.M., A Heuristic Program for Assembly Line Balancing, Prentice-Hall, Englewood Cliffs, New Jersey, 1961.

36. Tonge, F.M., "Assembly Line Balancing using Probabilistic Combinations of Heuristics," <u>Management Science</u>, Vol. 11, No. 7, May, 1965, p 727.

37. Thomopoulus, N.J., "Line Balancing Sequencing for Model Mix Assembly", <u>Management Science</u>, Vol.14, No.2, Oct., 1967, p B-59.

38. Thomopoulus, N.J., "Mixed Model Line Balancing with Smoothed Station Assignments", <u>Management Science</u>, Vol. 16, No.9, May, 1970, p 593.

APPENDIX A

ESTIMATION OF THE PARAMETER OF AN EXPONENTIALLY DISTRIBUTED FUNCTION

The parameter of exponentially distributed set of emperical points is estimated by using a mathematical method called 'Curve Fitting'. A general exponential relationship between the two variables x and y may be established in the form

$$y = \beta_0 e^{\beta_1 x}$$

where β_0 and β_1 are parameters of the relationship between x and y. This relationship can also be expressed as

$$\log y = \log \beta_0 + \beta_1 x$$

After fitting the linear curve between x and log y, the linear coefficient β_1 and log β_0 are computed using the following expressions, which are established by applying least-square method.

$$\beta_1 = \frac{\sum_{i=1}^{N} x_i \sum_{i=1}^{N} \log y_i - N \sum_{i=1}^{N} x_i \log y_i}{\log e \left[\left(\sum_{i=1}^{N} x_i \right)^2 - N \sum_{i=1}^{N} x_i^2 \right]}$$

$$\log \beta_0 = \frac{1}{N} \left[\sum_{i=1}^{N} \log y_i - \beta_1 \log e \sum_{i=1}^{N} x_i \right]$$

where log e = 0.4343, and

x_i, y_i = emperical points in the set, i = 1,2,...,N

APPENDIX B

COMPUTER PROGRAM LISTING

```
          ***********************************
          *                                 *
          *    PROGRAM LISTING OF PHASE I   *
          *                                 *
          ***********************************


----FOLLOWING IS INPUT TO THE PROGRAM

----M IS EQUAL TO TOTAL NO. OF TASKS IN THE ASSEMBLY LINE
----NOPRTR IS TOTAL NO. OF OPERATORS AVAILABLE
----FOLOW(I,J) IS IMMEDIATE FOLLOWER MATRIX
----NT(I) IS TOTAL NO. OF TASKS FOLLOWING TASK (I)
----N IS MAXIMUM OF NT(I),S
----NC1 INITIAL CYCLE TIME TAKEN
----AVE(I) IS AVERAGE NORMAL TASK TIME FOR TASK (I)

----FOLLOWING ARE THE OUTPUT DATA

----KKK(I) SEQUENCE OF TASKS GENERATED
----ISLAK(I) IS THE SLAK TIME LEFT AT STATION (I)
----JBASIN(I) NUMBER OF TASKS ASSIGNED TO STATION (I)
----BD(I) IS THE BALANCE DELAY FOR SEQUENCE (I)
----MOPR(I) IS NO. OF STATIONS NEEDED FOR SEQUENCE (I)

---OTHER IMPORTANT VARIABLES

---LFTIM(I) TIME LEFT TO BE ASSIGNED TO THE STATION (I)
---LISTA,LISTB,FITLST--ARE LISTS AS DISCUSSED IN THE PAPER
---WT1,WT2,WT3,WT4,WT,REALWT ARE WEIGHTS AS STATED INTHE PAPER


   COMMON/LAB1/NSET,NT
   COMMON/LAB2/LFTIM,LISTA,LISTB,ISLAK,KKK,NOFOLW,JR,KONTER
   COMMON/LAB3/WT1,WT2,WT3,WT4,WT,REALWT,TOTAL,AB
   COMMON/LAB4/FOLOW,AVE,FITLST,M,N,NOPRTR,OPR,C,NTOT
   DIMENSION JT(100,100),NSET(100),JT1(100),NT(99),NSPT(100)
   DIMENSION LFTIM(30),LISTA(50),LISTB(50),ISLAK(30),KKK(100)
   DIMENSION NOFOLW(100),JR(100),KONTER(100),WT1(50),WT2(50),WT3(50)
   DIMENSION WT4(50),WT(50),REALWT(50),TOTAL(50),AB(50)
   DIMENSION SINDX(100),MCPR(200),JBASIN(30)
   DIMENSION BD(200),NEQSX(200),NTSK(5),NAVE(100)
   INTEGER FOLOW(100,20),AVE(100),FITLST(100),NP(100)
   INTEGER C(60),OPR
   READ 1,M,N,NOPT
1  FORMAT(3I3)
   DO 2 I=1,M
 2 READ 3,(FOLOW(I,J),J=1,N)
 3 FORMAT(40I3)
   READ 9,(NT(I),I=1,M)
   PRINT 6,M
   DO 7 I=1,M
 7 PRINT 8,I,(FOLOW(I,J),J=1,N)
```

```
 8 FORMAT(6X,I3,10X,30I3)
9  FORMAT(40I2)
 6 FORMAT(2X,*TOTAL NO, OF TASKS IN THE ASSEMBLY =*,I3,///,6X,*TASK*,
  11 X,*IMMEDIATELY FOLLOWING TASKS*)
   READ 4,(AVE(I1),I1=1,M)
   READ 59,MINST
4  FORMAT(20I4)
9  FORMAT(I3)
---MINST IS THE MINIMUN STEP LENGTH THAT CAN BE TAKEN
   PRINT 4444
44 FORMAT(4X,*TASK NO.*,6X,*AVERAGE TASK TIME*)
   DO 4445 I1=1,M
45 PRINT 4446,I1,AVE(I1)
46 FORMAT(6X,I3,15X,I3)
---DETERMINATION OF FOLLOWER MATRIX
   DO 50 I=1,M
   DO 50 J=1,M
0  JT(I,J)=0
   DO 1500 I=1,M
   NK=NT(I)
   IF(NK.NE.0) GO TO 4447
   GO TO 1500
447 CONTINUE
   KM1=1
----JT(I,J) IS THE IMMEDIATELY FOLLOWIER MATRIX. IT IS A BINARY MATRIX.
    JT(I,J)=1 IMPLIES THAT J FOLLOWS I IMMEDIATELY. JT(I,J)=0 IMPLIES
    THAT J DOES NOT FOLLOW I IMMEDIATELY.
    DO 1502 K=1,NK
    DO  1501 J=KM1,M
    IF(J.EQ.FOLOW(I,K)) GO TO 1503
    GO TO 1501
503 JT(I,J)=1
    GO TO 1502
501 CONTINUE
502 CONTINUE
500 CONTINUE
    PRINT 1505
505 FORMAT(//,1X,* FOLLOWING IS IMMEDIATELY FOLLOWER MATRIX*,//)
    DO 1506 I=1,M
506 PRINT 1507,(JT(I,J),J=1,M)
507 FORMAT(60I2)
-----DETERMINATION OF TOTAL FOLLOWER MATRIX
-----SUBROUTINE TFROMN GIVES THE TOTAL FOLLOWER MATRIX
     CALL TFROMN(M,JT)
     PRINT 1510
1510 FORMAT(//,1X,* FOLLOWING IS TOTAL FOLLOWER MATRIX*,//)
-----HERE JT(I,J) BECOMES TOTAL FOLLOWER MATRIX. IT IS AGAIN A BINARY MATRI
     JT(I,J)=1 IMPLIES THAT J FOLLOWS I AND JT(I,J)=0 IMPLIES THAT J DOES N
     FOLLOW I
     DO 1511 I=1,M
1511 PRINT 1512,(JT(I,J),J=1,M)
1512 FORMAT(60I2)
-----NOFOLW(I) IS TOTAL NO. OF TASKS FOLLOWING TASK  (I)
```

```
      DO 1513 I=1,M
      J1=0
      DO 1514 J=1,M
      IF(JT(I,J).NE.0) J1=J1+1
 1514 CONTINUE
 1513 NOFOLW(I)=J1-1
      DO 111 I=1,M
  111 PRINT 1515,I,NOFOLW(I)
 1515 FORMAT(1X,*TOTAL NO. OF TASKS FOLLOWING TASK(*,I2,*)=*,I2)
      READ 101,NSTEP,NC1
  101 FORMAT(I2,I4)
      IP=1
C-----IP IS AN INDEX TO SHOW THE CYCLE NO. BEING CONSIDERED
      C(IP)=NC1
  102 CONTINUE
      PRINT 115,C(IP)
  115 FORMAT(//,1X,*CYCLE TIME BEING CONSIDERED *,I4,//)
      NOPRTR=NOPT
      JP1=0
      * JP1 * COUNTER TO RECCRD NO. OF TASKS EXCEEDING CTCLE TIME
      DO 104 IP1=1,M
      IF(AVE(IP1).GT.C(IP)) GO TO 103
      GO TO 104
  103 JP1=JP1+1
      JIG1=AVE(IP1)
      JIG2=C(IP)
      JIG3=JIG1/JIG2
      NSPT(JP1)=JIG3+1
      NOPRTR=NOPRTR-1
      NP(JP1)=NSPT(JP1)*C(IP)-AVE(IP1)
      NTSK(JP1)=IP1
      NAVE(IP1)=AVE(IP1)
      AVE(IP1)=C(IP)-NP(JP1)
      PRINT 116,JIG3,IP1
  116 FORMAT(1X,I3,*PARALLEL STATIONS ARE REQD. FOR TASK NO.*,I3)
  104 CONTINUE
      IF(JP1.EQ.0.AND.C(IP).NE.NC1) GO TO 1112
C-----DETERMINATION OF SUM OF ALL TASK TIMES FOLLOWING EACH TASK
      DO 1015 I=1,M
      IISOM=0
      DO 1016 J=1,M
      IF(J.EQ.I) GO TO 1016
      IF(JT(I,J).GT.0) IISOM=IISOM+AVE(J)
 1016 CONTINUE
 1015 JR(I)=IISOM
      DO 1018 I=1,M
 1018 PRINT 1019,I,JR(I)
 1019 FORMAT(1X,*SUM OF ALL TASK TIMES FOLLOWING TASK(*,I2,*)=*,I4)
 1112 CONTINUE
      NTOT=0
      DO 1111 I=1,M
 1111 NTOT=NTOT+AVE(I)
      NCMIN=NTOT/NOPRTR+1
```

```
      IF(JP1.GT.2) GO TO 112
      IF(C(IP)*NOPRTR.EQ.NTOT) GO TO 114
      IF(C(IP).LT.NCMIN) GO TO 112
  114 CONTINUE
C-----BALANCE THE LINE FOR THIS CYCLE TIME,C(IP), USING COMSOAL
      CALL COMSOL(IP)
      IF(OPR.GT.NOPRTR) GO TO 131
      IF(OPR.EQ.NOPRTR) GO TO 106
      IF(OPR.LT.NOPRTR) GO TO 108
 106  IP=IP+1
      C(IP)=C(IP-1)-NSTEP
      IF(JP1.EQ.0) GO TO 102
      DO 117 IK1=1,JP1
      ITS=NTSK(IK1)
  117 AVE(ITS)=NAVE(ITS)
      GO TO 102
 108  IP=IP+1
      NNTEP=2*NSTEP
      C(IP)=C(IP-1)-NNTEP
      IF(C(IP).GT.NCMIN) GO TO 119
 1119 CONTINUE
      NNTEP=NNTEP/2
      IF(NNTEP.LT.MINST) GO TO 112
      C(IP)=C(IP)+NNTEP
      IF(C(IP).GT.NCMIN) GO TO 119
      GO TO 1119
      IF(JP1.EQ.0) GO TO 102
      DO 118 IK1=1,JP1
      ITS=NTSK(IK1)
  118 AVE(ITS)=NAVE(ITS)
      GO TO 102
 119  CONTINUE
      IF(JP1.EQ.0) GO TO 102
      DO 120 IK1=1,JP1
      ITS=NTSK(IK1)
  120 AVE(ITS)=NAVE(ITS)
      GO TO 102
 112  PRINT 113
 113  FORMAT(1X,*LAST SOLUTION WAS OPTIMAL*)
      GO TO 132
 131  NNTEP=NSTEP/2
      IF(NNSTEP.LT.MINST) GO TO 112
      IP=IP+1
      C(IP)=C(IP-1)+NNTEP
      GO TO 102
 132  CONTINUE
      STOP
      END
```

```
      **************************************************************
      * COMSOL IS THE MAIN SUBROUTINE WHICH BALANCES THE LINE *
      * FOR A GIVEN CYCLE TIME C(MC1)                          *
      **************************************************************

      SUBROUTINE COMSOL(MC1)
      COMMON/LAB1/NSET,NT
      COMMON/LAB2/LFTIM,LISTA,LISTB,ISLAK,KKK,NOFOLW,JR,KONTER
      COMMON/LAB3/WT1,WT2,WT3,WT4,WT,REALWT,TOTAL,AB
       COMMON/LAB4/FOLOW,AVE,FITLST,M,N,NOPRTR,OPR,C,NTOT
      DIMENSION NT(99),NSPT(100)
      DIMENSION LFTIM(30),LISTA(50),LISTB(50),ISLAK(30),KKK(100)
      DIMENSION NOFOLW(100),JR(100),KONTER(100),WT1(50),WT2(50),WT3(50)
      DIMENSION WT4(50),WT(50),REALWT(50),TOTAL(50),AB(50)
      DIMENSION SINDX(100),MCPR(200),JBASIN(30)
      DIMENSION BD(200),NEQSX(200),NTSK(5),NAVE(100)
      INTEGER FOLOW(100,20),AVE(100),FITLST(100),NP(100)
      INTEGER C(60),OPR
      MLA=50
      DO 3003 NUMBER=1,MLA
      SINDX(NUMBER)=10000.0
C---FORMATION OF LIST-A
      DO 100 K=1,M
      IJ=0
      DO 1020 I=1,M
      NK=NT(I)
      IF(NK.EQ.0) GO TO 1020
      DO 10 J=1,NK
   10 IF(FOLOW(I,J).EQ.K) IJ=IJ+1
 1020 CONTINUE
  100 LISTA(K)=IJ
C----OPR IS THE OPERATOR NO. WHICH IS CURRENTLY BEING ASSIGED TO TASKS
C----MLEFT IS THE NO. OF TASKS ASSIGNED TO OPR
C----ASSIGNMENT OF MLEFT TASKS TO THE OPERATOR (OPR)
      MN=1
      OPR=1
      MLEFT=1
      ISUMM=0
      LFTIM(OPR)=C(MC1)
 3000 CONTINUE
      K=0
C-----FORMATION OF LIST-B
      DO 15 I=1,M
      IF(LISTA(I).EQ.0) GO TO 16
      GO TO 15
   16 K=K+1
      LISTB(K)=I
   15 CONTINUE
      KK=K
C-----FORMATION OF FITLIST-C
C-----KK IS THE TOTAL NO. OF TASKS IN LIST-B
 2103 CONTINUE
```

```
      K3=1
      DO 21  IK=1,KK
      K2=LISTB(IK)
      IF(AVE(K2).LE.LFTIM(OPR))GO TO 22
      GO TO 21
   22 FITLST(K3)=K2
      K3=K3+1
   21 CONTINUE
      K4=K3-1
      ISLAK(OPR)=C(MC1)-ISUMM
      JBASIN(OPR)=MLEFT-1
      IF(K4.EQ.0) GO TO 2100
      GO TO 2000
  100 OPR=OPR+1
      IF(OPR.LE.NOPRTR) GO TC 2102
      MOPR(NUMBER)=OPR
      GO TO 3003
  102 MLEFT=1
      ISUMM=0
      LFTIM(OPR)=C(MC1)
      GO TO 2103
  000 IF(K4.EQ.1) KKK(MN)=FITLST(K4)
      IF(K4.EQ.1)GO TO 55
----K4 IS THE TOTAL NO. OF TASKS IN FITLIST-C
----ASSIGNING WEIGHTAGES TO ELEMENTS OF FITLST-C
      ISUM1=0
      DO 25 IIK=1,K4
      I3=FITLST(IIK)
   25 ISUM1=ISUM1+AVE(I3)
      DO 26 IIK=1,K4
      I2=FITLST(IIK)
   26 WT1(IIK)=FLOAT(AVE(I2))/FLOAT(ISUM1)
-----IU IS THE NO. OF TASKS UNASSIGED SO FAR
      IU=M-MN
      SUM2=0.0
      DO 27 IIK=1,K4
      I4=FITLST(IIK)
      AB(I4)=1.0/FLOAT(IU-NOFOLW(I4))
   27 SUM2=SUM2+AB(I4)
      DO 28 IIK=1,K4
      I5=FITLST(IIK)
   28 WT2(IIK)=AB(I5)/SUM2
      ISUM3=0
      DO 29 IIK=1,K4
      I6=FITLST(IIK)
   29 ISUM3=ISUM3+NOFOLW(I6)+1
      DO 30 IIK=1,K4
      I7=FITLST(IIK)
   30 WT3(IIK)=FLOAT(NOFOLW(I7)+1)/FLOAT(ISUM3)
      ISUM4=0
      DO 31 IIK=1,K4
      I8=FITLST(IIK)
   31 ISUM4=ISUM4+AVE(I8)+JR(I8)
```

```
      DO 32 IIK=1,K4
      I9=FITLST(IIK)
   32 WT4(IIK)=FLOAT(AVE(I9)+JR(I8))/FLOAT(ISUM4)
C-----DECISION OF FINAL WEIGHTAGES
      DO 40 IIK=1,K4
   40 WT(IIK)=WT1(IIK)*WT2(IIK)*WT3(IIK)*WT4(IIK)
      SUM6=0.0
      DO 41 IIK=1,K4
   41 SUM6=SUM6+WT(IIK)
      DO 42 IIK=1,K4
   42 REALWT(IIK)=WT(IIK)/SUM6
      SUM7=0.0
      DO 43 IIK=1,K4
      SUM7=SUM7+REALWT(IIK)
   43 TOTAL(IIK)=SUM7
      X=RNDY1(DUMMY)
      DO90J5=1,K4
      IF(X.LE.TOTAL(1)) GO TO 51
      IF(X.LE.TOTAL(J5+1).AND.X.GT.TOTAL(J5)) GO TO 52
   90 CONTINUE
   51 KKK(MN)=FITLST(1)
      GO TO 53
   52 KKK(MN)=FITLST(J5+1)
   53 CONTINUE
      IF((J5+1).EQ.K4)GO TO 54
      GO TO 55
   54 K4=J5
      GO TO 55
   55 CONTINUE
      II2=KKK(MN)
      DO 70 J=1,N
      IF(FOLOW(II2,J).EQ.0) GO TO 71
   70 KONTER(J)=FOLOW(II2,J)
   71 JK1=J-1
      IF(JK1.EQ.0) GO TO 80
      DO 72 JK2=1,M
      IF(JK2.EQ.KKK(MN)) GO TO 76
      GO TO 77
   76 LISTA(JK2)=100
      GO TO 72
   77 CONTINUE
      DO 73 J=1,JK1
   73 IF(JK2.EQ.KONTER(J)) LISTA(JK2)=LISTA(JK2)-1
   72 CONTINUE
      GO TO 81
   80 DO 82 JK2=1,M
      IF(JK2.EQ.KKK(MN)) LISTA(JK2)=100
   82 CONTINUE
   81 CONTINUE
      MLEFT=MLEFT+1
      II1=KKK(MN)
      LFTIM(OPR)=LFTIM(OPR)-AVE(II1)
      ISUMM=ISUMM+AVE(II1)
```

```
      MN=MN+1
      IF(MN.LE.M) GO TO 3
      JBASIN(OPR)=MLEFT-1
      ISLAK(OPR)=C(MC1)-ISUMM
C-----DETERMINATION OF SMOOTHNESS INDEX
      IPP=
      DO 84  ISL=1,OPR
   84 IPP=IPP+ISLAK(ISL)
      IPAV=IPP/OPR
      ISLK=
      DO 94  ISL=1,OPR
   94 ISLK=ISLK+(ISLAK(ISL)-IPAV)**2
      SMOTH=FLOAT(ISLK)
      SINDX(NUMBER)=SQRT(SMOTH)
      PRINT 2003,(KKK(I),I=1,M)
 2003 FORMAT(40I3)
      PRINT 3011,SINDX(NUMBER),(JBASIN(I),ISLAK(I),I=1,OPR)
 3011 FORMAT(1X,*SMOOTH INDEX =*,F9.2,*TASKS ASSIGNED..SLAK TIMES*,12(2I
     13,1X))
C-----CALCULATION OF BALANCE DELAY
      MOPR(NUMBER)=OPR
      NDUM=OPR*C(MC1)
      NNNN=NDUM-NTOT
      BD(NUMBER)=FLOAT(NNNN)/FLOAT(NDUM)
      IF(NUMBER.EQ.1) GO TO 93
      DIFER=BD(NUMBER)-BD(NUMBER-1)
      IF(DIFER.LE.0.0001) GO TO 91
   93 CONTINUE
      PRINT 107,BD(NUMBER)
  107 FORMAT(1X,*BALANCE DELAY=*,F8.4)
   91 CONTINUE
 3003 CONTINUE
      SMALL=
      DO 101 NRT=1,MLA
      IF(SINDX(NRT).LT.SMAL) GO TO 102
      GO TO 101
  102 SMAL=SINDX(NRT)
      KRT=NRT
  101 CONTINUE
      NCTR=
      DO 103 NSUB=1,MLA
      DIFR=SINDX(NSUB)-SINDX(KRT)
      IF(ABS(DIFR).LE.4.0) GO TO 104
      GO TO 103
  104 NCTR=NCTR+1
      NEQSX(NCTR)=NSUB
      IF(NCTR.EQ.0) GO TO 3004
  103 CONTINUE
      PRINT 105,SINDX(KRT),(NEQSX(NOTR),NOTR=1,NCTR)
  105 FORMAT(//1X,*MIN SMOOTHNESS INDEX=*,F10.2,*FOLLOWING SEQUENCE NOS.
     1 ARE OPTIMAL FOR THIS CYCLE TIME *,//,50I3)
 3004 CONTINUE
      CALL MIN(MLA,MOPR,IIII)
```

```
      OPR=MOPR(IIII)
      PRINT 3005,OPR,IIII
 3005 FORMAT(1X,*MIN. NO. OF OPR.  *,I3,*FOR SEQUENCE *,I3)
      RETURN
      END
```

```
*****************************************************************
* TFROMN SUBROUTINE TRANSFORMS IMMEDIATELY FOLLOWER BINARY *
* MATRIX,JT(I,J), INTO TOTAL FOLLOWER MATRIX JT(I,J)        *
*****************************************************************
```

```
      SUBROUTINE TFROMN(M,JT)
      DIMENSION MSET(100),JT1(100),JT(100,100)
      DO 10 J=1,M
   10 MSET(J)=0
      DO 2 J=1,M
      DO 3 I=1,M
      IF(JT(I,J).NE.0) GO TO 22
      GO TO 3
   22 MSET(J)=MSET(J)+1
    3 CONTINUE
    2 CONTINUE
      DO 4J=1,M
      IF(MSET(J).EQ.0) GO TO 23
      GO TO 4
   23 DO 5 I=1,M
      JT(I,J)=0
    5 IF(I.EQ.J) JT(I,J)=1
    4 CONTINUE
      DO 6 J=1,M
      IF(MSET(J).NE.0) GO TO 24
      GO TO 6
   24 CONTINUE
      DO 7 I=1,M
    7 JT1(I)=JT(I,J)
      KJ1=J-1
      ISUM=0
      DO 8 I=1,KJ1
      DO 9 K=1,KJ1
    9 ISUM=ISUM+JT1(K)*JT(I,K)
      JT(I,J)=ISUM
      ISUM=0
    8 CONTINUE
      JT(J,J)=1
    6 CONTINUE
      DO 50 I=1,M
      DO 50 J=1,M
   50 IF(JT(I,J).GT.0) JT(I,J)=1
      RETURN
```

```
      END



      ************************************************************
      * SUBROUTINE MINIMA GIVES THE FIRST MINIMUN ELEMENT K OF    *
      * THE ARRAY NOY(I).                                         *
      ************************************************************

      SUBROUTINE MIN(N,NOY,K)
      DIMENSION NOY(100)
      LES=1000
      DO 10 JK=1,N
      IF(NOY(JK).LT.LES) GO TO 2
      GO TO 10
    2 LES=NOY(JK)
      K=JK
   10 CONTINUE
      RETURN
      END
```

```
          ***********************************
          *                                 *
          *  PROGRAM LISTING FOR PHASE II   *
          *                                 *
          ***********************************

          *********************************************
          * THE MAIN PROGRAM OPTIMIZES THE BUFFER LEVELS *
          *         USING UNIVARIATE TECHNIQUE          *
          *********************************************

C      FOLLOWING ARE THE INPUT DATA TO THE SYSTEM
C
C-----N IS TOTAL NO. OF WORK STATIONS IN THE ASSEMBLY
C-----SIG(I) IS THE STANDARD DEVIATION OF (I)TH WORK STATION TIME
C-----FEXP(I) IS THE PARAMETER OF NEGATIVE EXPONENTIALLY DISTRIBUTED
      INTER-FAILURE TIME FOR (I)TH WORK STATION
C-----MUREP(I),SIGREP(I) ARE THE MEAN AND STANDARD DEVIATION OF REPAIR
      TIME DISTRIBUTION FOR (I)TH WORK STATION
C-----NLL(I) IS THE NO. OF BUFFER SPACES IN THE (I)TH GROUP OF VI
C-----NJ1(I,J) IS THE  (I)TH MEMBER BUFFER SPACE OF (J)TH GROUP OF VI
C-----MMM IS THE NUMBER OF ITEMS PRODUCED,AFTER WHICH OUTPUT RATE IS
      CALCULATED
C
C      FOLLOWING ARE THE DATA RELATED TO THE COSTS OF THE SYSTEM
C
C-----ALAB IS THE FIXED LABOUR COST PER SHIFT PER LABOUR
C-----CST IS THE LABOUR COST OF PRODUCING AN ADDITIONAL ITEM (THIS IS
      FOR INCENTIVE PLAN)
C-----SOUT IS THE OUTPUT RATE OVER WHICH THE INCENTIVE IS PAID
C-----C3(I) IS THE COST OF HOLDING BUFFER PER UNIT, AT (I)TH BUFFER STATI
C-----C1 IS THE MATERIAL COST PER ITEM
C-----OVVR IS THE FIXED OVERHEAD COST PER SHIFT
C-----OVRHD1 IS THE VARIABLE OVERHEAD COST PER SHIFT
C
C      OTHER IMPORTANT VARIABLES
C
C-----TNARV(I) IS,THE TIME OF NEXT ARRIVAL IN (I)TH WORK STATION
C-----TNDPR(I) IS THE TIME OF THE NEXT DEPARTURE FROM (I)TH STATION
C-----NFALTM(I) IS THE TIME AT WHICH (I)TH WORK STATION FAILS
C-----QUE(I) IS THE VALUE EQUAL TO THE LENGTH OF THE QUEUE AFTER (I)TH
      WORK STATION (I. E.,NUMBER OF ITEMS IN BUFFER (I) )
C-----STATUS(I) INDICATES THE STATUS OF THE (I)TH WORK STATION
C-----ATIME INDICATES THE TIME OF THE MOST RECENT CHANGEIN THE STATUS
      OF THE WORK STATIONS
C-----CUSERV(I) CONTAINS THE NO. OF UNITS PRODUCED BY (I)TH STATION TILL
      ATIME
C-----CUMQUE(I,J) HOLDS THE UNITS OF TIME THE BUFFER (I-1) HAS HAD (J-1)
C      ITEMS IN IT SINCE THE STARTING TIME OF THE PROCESS
C------CUMUTL(I) IS THE TOTAL PRODUCTIVE TIME OF (I)TH STATION TILL ATIM
C-----IST(I) IS THE ACTUAL TIME TAKEN BY THE (I)TH STATION TO COMPLETE
```

```
C      ONE ITEM
C-----NPRARR(I) HOLDS THE VALUE OF THE TIME OF THE PREVIOUS ARRIVAL IN
C      THE (I)TH WORK STATION
C-----NPRDPR(I) HOLDS THE VALUE OF THE TIME OF THE PREVIOUS DEPARTURE
C      FROM THE (I)TH WORK STATION
C
C      FOLLOWING ARE THE DESIGN VARIABLES
C
C-----BFR(I) CONTAINS THE VALUE OF THE BUFFER CAPACITY FOR (I)TH GROUP
C      OF VARB
C-----BBB(I) IS THE BUFFER CAPACITY BETWEEN (I)TH AND (I+1)TH WORK STATION


      COMMON/OUT1/C1,SOUT,SHIFT,ALAB,CST,OVVR,PRESN,FACTOR,C3(30)
      COMMON/SFT/MXX,MNO,MMM,MX
      COMMON/PREARR/NPRARR(30),NPRDPR(30)
      COMMON/LAB1/QUE,CUMQUE,TIME,TNARV,TNDPR,STATUS,ATIME
      COMMON/LAB2/CUMUTL,CUSERV
      COMMON/LAB3/MU,SIG
      COMMON/FAIL/FEXP,SIGREF,MUREP,NFALTM,NARV
      INTEGER BFR,BBB,T,ATIME,STATUS(30),B(30),CUMUTL(30),CUSERV(30)
      INTEGER QUE(30),CUMQUE(30,30),TIME(30),TNARV(30),TNDPR(30)
      DIMENSION BFR(30),NSQ(30),NLL(30),NJ1(30,30),BBB(30),NFALTM(300)
      DIMENSION(300),NRT(300),NARV(30),ACUMQ(30),MU(30),SIG(30),IST(30),
     1T(150),Z(150),PCUTIL(30),OUTRAT(100),MUREP(30),SIGREP(30),FEXP(30)
C-----READ THE INPUT DATA
      READ 110,NL,NSTEP,N
 110  FORMAT(10I7)
      READ 111,(NLL(I),I=1,NL)
 111  FORMAT(14I3)
      DO 112 I=1,NL
      J=NLL(I)
 112  READ113,(NJ1(I,K),K=1,J)
 113  FORMAT(14I3)
      N6=N-1
      READ114,SHIFT,C1,SOUTALAB,OVRHD,CST,OVVR,FACTOR
 114  FORMAT(8F10.2)
      READ 115,(C3(I),I=1,N6)
 115  FORMAT(14F6.2)
      DO 8,I=1,N
   8  READ30,MUREP(2),SIGREP(2),FEXP(2)
  30  FORMAT(I4,2F8.1)
      READ117,PRESN
 117  FORMAT(F10.4)
      READ 2,(MU(I),I=1,N)
   2  FORMAT(20I4)
      READ 3,(SIG(I),I=1,N)
   3  FORMAT(20F6.1)
      READ5,MMMZMX,MXX,MNO
   5  FORMAT(4I4)
C-----CHOOSE A STARTING POINT BFR
      READ 308,(BFR(I),I=1,NL)
 308  FORMAT(10I3)
```

```
C-----FIND THE FUNCTION VALUE FNQ AT THE STARTING POINT
      CALL BUFER(NL,NLL,NJ1,BFR,BBB)
      CALL MAIN(N6,BBB,FNQ)
      PRINT 100,FNQ,(BFR(I),I=1,NL)
 100  FORMAT(1X,*INITIAL OB-FN VALUE =*,F10.2,/1X,*INITIAL DESIGN VECTOR
     1 =*,20I4)
      NQ=1
      K=0
    9 CONTINUE
      J=0
C-----CHOOSE THE DIRECTION OF MOVEMENT NSQ
      DO 10 I=1,NL
      IF(I.EQ.NQ) GO TO 31
      NSQ(I)=0
      GO TO 10
   31 NSQ(I)=1
   10 CONTINUE
      NT=NSTEP
C-----SEARCH IN THE CHOSEN DIRECTION TILL THE FUNCTION VALUE DECREASES
   14 CONTINUE
      DO 11 I=1,NL
      BFR(I)=BFR(I)+NT*NSQ(I)/2
      IF(BFR(I).LE.1) GO TO 201
      GO TO 11
  201 BFR(I)=BFR(I)-NT*NSQ(I)/2
      IZERO=1
      GO TO 13
   11 CONTINUE
C-----FIND THE FUNCTION VALUE FNT AT THE NEW POINT BFR
      CALL BUFER(NL,NLL,NJ1,BFR,BBB)
      CALL MAIN(N6,BBB,FNT)
      PRINT 8708,FNT,(BFR(I),I=1,NL)
 8708 FORMAT(1X,*FNT= *,F10.3,*BFR  = *,20I3)
      IF(FNT.LT.FNQ) GO TO 12
C-----START SEARCHING IN ANOFTER DIRECTION NSQ
      GO TO 13
   12 J=1
      FNQ=FNT
      PRINT 101,FNQ,(BFR(I),I=1,NL)
  101 FORMAT(1X,*NEXT MIN. VALUE OF OB-FN '=*,F10.2,/1X,*NEXT DE,+P -PF9
     1NT =*,20I4)
      NT=2*NT
      GO TO 14
   13 IF(J.EQ.0) GO TO 15
      K=1
      IF(IZERO.EQ.1) GO TO 202
      DO 16 I=1,NL
   16 BFR(I)=BFR(I)-NT*NSQ(I)/2
  202 CONTINUE
   15 NQ=NQ+1
      IF(NQ.GT.NL) GO TO 17
      GO TO 9
   17 IF(K.EQ.0) GO TO 25
```

```
      K=
      NO=
      GO TO 9
 25   PRINT26,FNO,(BFR(I),I=1,NL)
 26   FORMAT( X,*MIN. FUNCTION VALUE IS '*,F10.3,/1X,*DESIGN VECTOR=*,2
     1I4)
      STOP
      END
```

```
                    *********************
                    *                   *
                    *  SUBROUTINE MAIN  *
                    *                   *
                    *********************

       ******************************************
       * THIS SUBROUTINE SIMULATES THE SYSTEM FOR *
       * THE GIVEN VALUE OF BUFFER INVENTORIES    *
       ******************************************
```

```
      SUBROUTINE MAIN(N6,B,FLNL)
      COMMON/OUT1/C1,SOUT,SHIFT,ALAB,CST,OVVR,PRESN,FACTOR,C3(30)
      COMMON/SFT/MXX,MNO,MMM,MX
      COMMON/PREARR/NPRARR(30),NPRDPR(30)
      COMMON/LAB1/QUE,CUMQUE,TIME,TNARV,TNDPR,STATUS,ATIME
      COMMON/LAB2/CUMUTL,CUSERV
      COMMON/SST/IST
      COMMON/LAB3/MU,SIG
      COMMON/LAB5/N
      COMMON/FAIL/FEXP,SIGREP,MUREP,NFALTM,NARV
      DIMENSION NFALTM(300),NFT(300),NRT(300),NARV(30),ACUMQ(30)
      DIMENSION MU(30),SIG(30),IST(30),OUTRAT(100),T(150),Z(150)
      DIMENSION PCUTIL(30),MUREP(30),SIGREP(30),FEXP(30)
      INTEGER QUE(30),CUMQUE(30,30),TIME(30),TNARV(30),TNDPR(30)
      INTEGER STATUS(30),B(30),CUMUTL(30),CUSERV(30),T,Z,ATIME
C-----INITIALIZATION OF THE 'SEED' FOR THE RANDOM NUMBERS SEQUENCES
      XXZZ=189277
      XXYY=186285
      Y=SNDY1(XXZZ)
      YY=SNDY4(XXYY)
      YYY=SNDY5(XXYY)
      N=N6+1
      N2=2*N
C-----FOLLOWING ARE THE INITIAL CONDITIONS FOR THE ASSEMBLY LINE
      IJKL=1
      ATIME=0
      JKA=1
      JKD=1
      T(IJKL)=0
      DO    I=1,N
```

```
      STATUS(N+1)=0
      TIME(I)=0
      STATUS(I)=
      CUSERV(I)=
      TNARV(I)=0
      NPRARR(I)=0
      CALL NORMAL(I,IST)
      TNDPR(I)=IST(I)
      NPRDPR(I)=0
      CUMUTL(I)=IST(I)
      IF(I.EQ.N) GO TO 1
      QUE(I)=B(I)/2
      IF(QUE(I).EQ.0) QUE(I)=1
      K1=B(I)+1
      L=I+1
      DO 2 J=1,K1
    2 CUMQUE(L,J)=0
    1 CONTINUE
      PRINT 20,(TNARV(I),I=1,N)
   20 FORMAT(1X,* NEXT ARRIVAL TIMES *,20I6)
      PRINT 21,(TNDPR(I),I=1,N)
   21 FORMAT(1X,* NEXT DEPARTURE TIMES *,20I6)
      QUE(N)=0
      N5=N-1
      NOPRT=1
C-----GENERATE THE INITIAL FAILURE TIMES FOR ALL THE WORK STATIONS
      DO 81 III=1,N
      CALL FAIL(III,FEXP,NFT)
   81 NFALTM(III)=TNARV(III)+NFT(III)
  100 CONTINUE
C-----CHECK IF ANY OF THE WORK STATIONS FAIL
      DO 82 III=1,N
      IF(TNARV(III).LT.NFALTM(III).AND.TNDPR(III).GE.NFALTM(III))GOTO82
      IF(TNDPR(III).LT.NFALTM(III)) GO TO 82
      IF(QUE(III-1).EQ.0) GO TO 42
      GO TO 43
C-----GENERATE THE REPAIR TIME FOR (III)TH WORK STATION
   42 CALL RPAIR(III,MUREP,SIGREP,NRT)
      NARV(III)=NPRARR(III)+NRT(III)
      TNARV(III)=MAX0(NARV(III),TNARV(III))
      TNDPR(III)=TNARV(III)+IST(III)
      STATUS(III)=0
      GO TO 44
   43 CALL RPAIR(III,MUREP,SIGREP,NRT)
      NARV(III)=MIN0(TNARV(III),TNDPR(III))+NRT(III)
      TNARV(III)=MAX0(TNARV(III),NARV(III))
      STATUS(III)=1
C-----GENERATE THE NEXT INTER FAILURE TIME FOR (III)TH WORK STATION
   44 CALL FAIL(III,FEXP,NFT)
      NFALTM(III)=TNARV(III)+NFT(III)
   82 CONTINUE
C-----FIND THE EARLIEST OCCURING EVENT
      DO 10 I=1,N
```

```
      K=2*I-1
      IF(STATUS(I).EQ.1) GO TO 3
      Z(K)=TNARV(I)
      Z(K+1)=99999
      GO TO 4
    3 Z(K)=99999
      Z(K+1)=TNDPR(I)
    4 CONTINUE
   10 CONTINUE
      CALL MIN(N2,Z,K9)
C-----CALL SUBROUTINE 'ARRIVE' OR 'DEPART' DEPENDING UPON WHICH EVENT,
C      ARRIVAL OR DEPARTURE, OCCURS FIRST
      DO 5 M1=1,N
      M2=2*M1
    5 IF(M2.EQ.K9) GO TO 6
      M3=(K9+1)/2
      CALL ARRIVE(M3)
      MM3=JKA*MMM
      IF(QUE(N).EQ.MM3) GO TO 62
      NOPRT=1
      GO TO 54
    6 CONTINUE
      CALL DEPART(M1,B)
      MM2=JKD*MMM
      IF(QUE(N).EQ.MM2) GO TO 61
      NOPRT=1
      GO TO 54
   61 JKD=JKD+1
      IF(NOPRT.EQ.1) GO TO 55
      GO TO 54
   55 PRINT 51,M1,Z(K9)
   51 FORMAT(1X,*DEPARTURE AT STATION  *,I2,*    AT TIME  =*,I7)
      NOPRT=NOPRT+1
      GO TO 8
   62 JKA=JKA+1
      IF(NOPRT.EQ.1) GO TO 7
      GO TO 54
    7 PRINT 50,M3,Z(K9)
   50 FORMAT(1X,*ARRIVAL AT STATION    *,I2,*    AT TIME  =*,I7)
      NOPRT=NOPRT+1
    8 CONTINUE
   54 CONTINUE
      MIJ=IJKL*MMM
      IF(QUE(N).LT.MIJ) GO TO 100
      T(IJKL+1)=ATIME
      PRINT 11,MIJ,T(IJKL+1)
   11 FORMAT(1X,*TIME TAKEN FOR PRODUCING*,I3,* SUB-ASSEMBLIES=*,I5)
      NOUT=QUE(N)
C-----COMPUTE THE OUTPUT RATE
      OUTRAT(IJKL+1)=FLOAT(NOUT)/FLOAT(ATIME)
      PRINT 60,OUTRAT(IJKL+1)
   60 FORMAT(1X,*OUTPUT RATE =*,F10.4)
      IF(IJKL.LT.MXX) GO TO 63
```

```
C-----TEST WHETHER THE RATE OF PRODUCTION HAS REACHED THE STEADY STATE
      AVOUT1=0.
      AVOUT2=0.
      DO 64 JJ=1,MNO
      JKK=IJKL-JJ
      AVOUT1=AVOUT1+OUTRAT(JKK)
   64 AVOUT2=AVOUT2+OUTRAT(JKK+1)
      AVOT1=AVOUT1/FLOAT(MNO)
      AVOT2=AVOUT2/FLOAT(MNO)
      RSHIO=ABS(AVOT1-AVOT2)/ABS(AVOT1)
      IF(RSHIO.GT.PRESN) GO TO 63
      GO TO 65
   63 IJKL=IJKL+1
      IF(IJKL.LE.MX) GO TO 100
   65 CONTINUE
C-----TEST THE STOCHASTIC CONVERGENCE OF THE SYSTEM
      AVV=AVOT2*SHIFT
      PRINT 66,AVV
      INDEX=0
      IF(INDEX.EQ.1) GO TO 6969
      STD1=0.
      DO 97 JJ=1,MNO
      JK1=IJKL-JJ
   97 STD1=STD1+(OUTRAT(JK1)-AVOT2))**2
      AMN=FLOAT(MNO-1)
      SIGM1=SQRT(STD1/AMN)
      SIGMS=SIGM1/SQRT(AMN)
      FIG=(SIG*TPROB/ACCUR)**2
      NDIFF=MNO-IFIX(FIG)
      IF(NDIFF.GT.0) GO TO 6870
      MADD=IJKL+NDIFF
      IJKL=IJKL+1
      INDEX=1
      IF(IJKL.LT.MADD) GO TO 100
 6870 CONTINUE
   66 FORMAT(///1X,*OUTPUT RATE PER SHIFT =*,F8.2,//)
C-----COMPUTE THE AVERAGE BUFFER LEVELS AT EACH BUFFER SPACE
      DO 25 NCUM=2,N
      NBISUM=0
      NBISM=0
      NBI=B(NCUM-1)+1
      DO 26 NSUM=1,NBI
      NBISUM=NBISUM+CUMQUE(NCUM,NSUM)
   26 NBISM=NBISM+NSUM*CUMQUE(NCUM,NSUM)
      ACUMQ(NCUM)=FLOAT(NBISM)/FLOAT(NBISUM)-1.0
   25 CONTINUE
      PRINT 27,(ACUMQ(NCUM),NCUM=2,N)
   27 FORMAT(1X,*AVERAGE BUFFERS = *,12F8.3)
C-----COMPUTE THE OBJECTIVE FUNCTION
      FON=0.0
      DO 30 JKE=2,N
   30 FON=FON+ACUMQ(JKE)*CB(JKE-1)
```

```
      PIN=1.7?*AVV-75.0
C-----THE GENERAL INTERPRETATION OF PIN IS THAT IT REPRESENTS THE INCREASE
C     PRODUCTION PER SHIFT, OVER WHICH THE INCENTIVE MONEY IS PAID. IN THE
C     TV ASSEMBLY LINE BEING CONSIDERED IT REPRESENTS THE INCREASED VALUE
C     OF 'PERFORMANCE LEVEL' PER SHIFT. PERFORMANCE LEVEL EQUAL TO 75.0 IS
C     CONSIDERED AS THE NORMAL PACE AT J.K.ELECTRONICS.
      IF(PIN.LE.0.0) GO TO 901
      ACNTV=PIN*CST*FLOAT(N)
      GO TO 902
  901 ACNTV=0.0
  902 CONTINUE
      AMATL=AVV*C1
      PLAB=ALAB*FLOAT(N)
      OVRHD1=FACTOR*(PLAB+ACNTV)
      FUN1=(FON+AMATL+PLAB+OVRHD1+ACNTV+OVVR)/AVV
      PRINT 31,FUN1
   31 FORMAT(//1X,*FUN1=*,F10.2)
C-----COMPUTE THE TOTAL UTILIZATION OF EACH WORK STATION
      DO 15 I=1,N
      NTIME=MAX0(TNARV(I),TNDPR(I))
   15 PCUTIL(I)=(FLOAT(CUMUTL(I))/FLOAT(NTIME))*100.0
      PRINT 16,(PCUTIL(I),I=1,N)
   16 FORMAT(1X,*UTILIZATION =*,//,20F8.2)
      RETURN
      END
```

```
                    *********************
                    *                   *
                    *  SUBROUTINE FAIL  *
                    *                   *
                    *********************

          **********************************************
          * THIS SUBROUTINE GENERATES THE NEXT         *
          * INTER FAILURE TIME FOR (I)TH WORK STATION *
          **********************************************
      SUBROUTINE FAIL(I,FEXP,NFT)
      DIMENSION NFT(300),FEXP(30)
    6 CONTINUE
      X=RNDY5(DUMMY)
      R3=-ALOG(1.0-X)*FEXP(I)
      NFT(I)=IFIX(R3)
      RETURN
      END
```

```
                    ***********************
                    *                     *
                    *  SUBROUTINE REPAIR  *
                    *                     *
```

```
                    **********************

                   ********************************
                   * THIS SUBROUTINE GENERATES THE *
                   * REPAIR TIME FOR (I)TH STATION *
                   ********************************

      SUBRCUTINL RPAIR(I,MU,SIG,NRT)
    6 CONTINUE
      DIMENSION NRT(300),MU(30),SIG(30)
      SUM=0.0
      DO 5 JK=1,6
      R=RNDY4(DUMY)
    5 SUM=SUM+R
      NRT(I)=IFIX(1.414*SIG(I)*(SUM-3.0))+MU(I)
      IF(NRT(I).LE.0) GO TO 6
      RETURN
      END

                    *******************
                    *                 *
                    * SUBROUTINE MIN  *
                    *                 *
                    *******************

          ******************************************
          * THIS SUBROUTINE GIVES THE MINIMUM VALUE *
          * OF THE ARRAY  Z                          *
          ******************************************

      SUBROUTINE MIN(N2,Z,K)
C-----IF THERE ARE TWO OR MORE MINIMAS IN THE ARRAY Z, THEN THIS SUBROUTI
C     WILL GIVE THE FIRST MINIMA OF THE SEQUENCE Z
      INTEGER Z(40)
      LES=1[illegible]
      DO 10 JK=1,N2
      IF(Z(JK).LT.LES) GO TO 2
      GO TO 10
    2 LES=Z(JK)
      K=JK
   10 CONTINUE
      RETURN
      END

                    **********************
                    *                    *
                    * SUBROUTINE NORMAL  *
```

```
                              *                        *
                              *************************
      SUBROUTINE NORMAL(L,IST)
      COMMON/LAB2/MU,SIG
      DIMENSION MU(30),SIG(30),IST(30)
      SUM=0.0
      DO 5 JK=1,6
      R=RANDYI(DUMY)
    5 SUM=SUM+R
      XX=(1.414*SIG(L))*(SUM-3.0)
      IST(L)=IFIX(XX)+MU(L)
      RETURN
      END
                              *************************
                              *                       *
                              *  SUBROUTINE ARRIVE    *
                              *                       *
                              *************************

         *************************************************************
         * THIS SUBROUTINE DOES ALL THE CHANGES IN THE SYSTEM WHICH  *
         * ARE DUE TO THE ARRIVAL OF AN ITEM IN THE (I)TH STATION    *
         *************************************************************


      SUBROUTINE ARRIVE(I)
      COMMON/PREARR/NPRARR(30),NPRDPR(30)
      COMMON/LAB1/QUE,CUMQUE,TIME,TNARV,TNDPR,STATUS,ATIME
      COMMON/SST/IST
      DIMENSION MU(30),SIG(30),IST(30)
      INTEGER QUE(30),CUMQUE(30,30),TIME(30),TNARV(30),TNDPR(30)
      INTEGER ATIME,STATUS(30)
C-----CHECK WHETHER THE WORK STATION UNDER CONSIDERATION IS THE FIRST ONE
C     IN THE ASSEMBLY LINE
      IF(I.EQ.1) GO TO 15
C-----CHECK IF THE BUFFER SPACE BEFORE THE STATION IS EMPTY
      IF(QUE(I-1).EQ.0) GO TO 26
      M=QUE(I-1)
      IOQ=M+1
      TIME(I)=TNARV(I)
C-----UPDATE CUMQUE,ATIME,QUE,STATUS,NPRARR
      ATIME=TIME(I)
      IF(ATIME.GE.NPRDPR(I-1)) GO TO 16
      CUMQUE(I,IOQ)=CUMQUE(I,IOQ)+ATIME-NPRARR(I)
      GO TO 17
   16 CUMQUE(I,IOQ)=CUMQUE(I,IOQ)+ATIME-MAXO(NPRDPR(I-1),NPRARR(I))
   17 NPRARR(I)=TNARV(I)
      QUE(I-1)=QUE(I-1)-1
C-----ESTABLISH NEXT ARRIVAL TIME AT (I)TH WORK STATION
      STATUS(I)=1
      TNARV(I)=TNDPR(I)
      GO TO 30
   26 CUMQUE(I,1)=CUMQUE(I,1)+TNARV(I)-NPRARR(I)
```

```
C-----WAIT TILL AN ITEM IS ADDED TO THE BUFFER SPACE BEFORE (I)TH WORK STA
      NPRARR(I)=TNARV(I)
      TIME(I)=TNARV(I)
      ATIME=TIME(I)
      TNARV(I)=TNDPR(I-1)+1
      TNDPR(I)=TNDPR(I-1)+IST(I)
      STATUS(I)=0
      GO TO 30
   15 TIME(I)=TNARV(I)
C-----UPDATE ATIME,STATUS
      ATIME=TIME(I)
      STATUS(I)=1
C-----ESTABLISH THE NEXT ARRIVAL TIME
      TNARV(I)=TNDPR(I)
   30 CONTINUE
      RETURN
      END
```

```
                         **********************
                         *                    *
                         * SUBROUTINE DEPART  *
                         *                    *
                         **********************

         ***********************************************************
         * THIS SUBROUTINE DOES ALL THE CHANGES IN THE SYSTEM WHICH *
         * ARE DUE TO DEPARTURE OF AN ITEM FROM THE WORK STATION    *
         ***********************************************************
```

```
      SUBROUTINE DEPART(I,B)
      COMMON/PREARR/NPRARR(30),NPRDPR(30)
      COMMON/LAB1/QUE,CUMQUE,TIME,TNARV,TNDPR,STATUS,ATIME
      COMMON/LAB3/CUMUTL,CUSERV
      COMMON/LAB2/MU,SIG
      COMMON/SST/IST
      COMMON/LAB5/N
      INTEGER ATIME,STATUS(30),B(30),CUMUTL(30),CUSERV(30)
      INTEGER QUE(30),CUMQUE(30,30),TIME(30),TNARV(30),TNDPR(30)
      DIMENSION MU(20),SIG(20),IST(20)
      IF(I.EQ.N) GO TO 45
      IF(QUE(I).GE.B(I)) GO TO 10
      M=QUE(I)
C-----UPDATE CUMQUE,ATIME,NPRDPR
      IOQ=M+1
      L=I+1
      CUMQUE(L,IOQ)=CUMQUE(L,IOQ)+TNDPR(I)-MAX0(NPRDPR(I),NPRARR(I+1))
   45 CONTINUE
      NPRDPR(I)=TNDPR(I)
```

```
      TIME(I)=TNDPR(I)
      ATIME=TIME(I)
      QUE(I)=QUE(I)+1
C-----THE WORK STATION IS NOW READY TO TAKE ONE ITEM FOR SERVICING
      STATUS(I)=
C-----ESTABLISH NEXT ARRIVAL AND DEPARTURE TIME OF AN ITEM FOR THIS STATIO
      TNARV(I)=MAX0(TNARV(I),TNDPR(I))
      CALL NORMAL(I,IST)
C-----UPDATE CUMUTL,CUSERV
      TNDPR(I)=TNARV(I)+IST(I)
      CUMUTL(I)=CUMUTL(I)+IST(I)
      CUSERV(I)=CUSERV(I)+1
      GO TO 4
   1  IF(TNARV(I).EQ.TNDPR(I)) GO TO 11
C-----SINCE NEXT BUFFER SPACE IS FULL WAIT TILL AN ITEM IS PICKED UP FROM
      TNARV(I)=MAX0(TNARV(I),TNARV(I+1))
      GO TO 12
  11  TNARV(I)=TNARV(I+1)
  12  TNDPR(I)=TNARV(I+1)+1
      STATUS(I)=1
  4   CONTINUE
      IF(I.EQ.N) GO TO 41
      IF(STATUS(I+1).EQ.1.AND.TNDPR(I).EQ.TNDPR(I+1))TNDPR(I+1)=TNDPR(I+
     11)-1
  41  CONTINUE
      RETURN
      END
```

```
                    *********************
                    *                   *
                    * SUBROUTINE BUFER  *
                    *                   *
                    *********************

        ***************************************************
        * THIS SUBROUTINE GROUPS THE BUFFER SPACES IN NL GROUPS *
        * OF V'S AS EXPLAINED IN SECTION 4.2.2                   *
        ***************************************************
```

```
      SUBROUTINE BUFER(NL,NLL,NJ1,BFR,BBB)
      INTEGER BFR,BBB
      DIMENSION NLL(30),NJ1(30,30),BBB(30),BFR(30)
      DO 10 I=1,NL
      J=NLL(I)
      DO 15 J1=1,J
      K=NJ1(I,J1)
  15  BBB(K)=BFR(I)
  10  CONTINUE
      RETURN
      END
```

```
              ****************************************
              *                                      *
              *   PROGRAM LISTING FOR CHI-SQUARE TEST  *
              *                                      *
              ****************************************
```

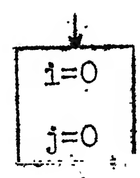
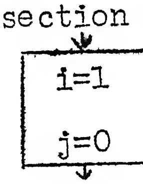
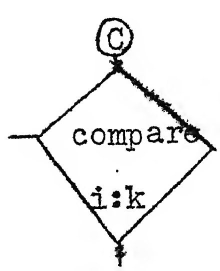
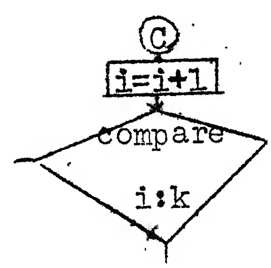
```
C-----XOBS IS THE ARRAY OF DATA UNDER TEST
      DIMENSION XOBS([illegible]),YOBS([illegible]),XEXP(8[illegible]),YEXP(8[illegible]),F(8[illegible]),AREA(8[illegible])
      DIMENSION Y([illegible]),XXOBS([illegible])
      INTEGER XOBS,YOBS
      READ 50,ALAM
   50 FORMAT(F[illegible].4)
      DO 9543 JKLF=1,54
C-----N IS THE TOTAL NUMBER OF DATA UNDER TEST
C-----K IS THE DEGREE OF FREEDOM FOR THE CHI-SQUARE TEST
      READ 100,N,K,L
  100 FORMAT(I2,2I4)
      READ 10 ,(XXOBS(I),I=1,N)
   10 FORMAT(16F5.[illegible])
      KM2=K-2
      KM1=K-1
C-----COMPUTE THE MEAN VALUE, AVMU, AND THE STANDARD DEVIATION VALUE,
C     OF THE DATA
      AMU=0.0
      DO 10 I=1,N
   10 AMU=AMU+XXOBS(I)
      AVMU=AMU/FLOAT(N)
      ASIG=0.0
      DO 11 I=1,N
   11 ASIG=ASIG+(XXOBS(I)-AVMU)**2
      SIG=SQRT(ASIG/FLOAT(N-1))
C-----DEFINE THE LOWER AND THE UPPER LIMIT OF THE VALUES OF THE DATA
C     DIVIDE THIS RANGE INTO (K-2) EQUAL PARTS.(I)TH PART OF THIS DIV
C     IS CALLED (I)TH GROUP
      SMALL=AVMU-3.0*SIG
      BIG=AVMU+3.0*SIG
      STEP=(BIG-SMALL)/FLOAT(KM2)
      PRINT 20,AVMU,SIG,SMALL,BIG,STEP
   20 FORMAT(1X,*MU=*,F8.2,*     SIG=*,F8.2,/1X,*LOWER LIMIT=*,F8.3,*
     XUPPER LIMIT=*,F8.3,*     STEP LENGTH=*,F8.3)
C-----COMPUTE XOBS(I), THE NUMBER OF DATA FALLING IN (I)TH GROUP MADE
      DO 12 I=1,K
   12 XOBS(I)=0
      DO 13 I=1,N
      AB=XXOBS(I)
      IF(AB.LT.SMALL) GO TO 15
      GO TO 16
   15 XOBS(1)=XOBS(1)+1
      GO TO 1[illegible]
   16 IF(AB.GT.BIG) GO TO 17
      DO 1[illegible] J=2,KM1
```

E

```
      BC1=SMALL+FLOAT(J-1)*STEP
      BC2=BC1+STEP
      IF(AB.GT.BC1.AND.AB.LT.BC2) GO TO 16
      GO TO 14
 16   XOBS(J)=XOBS(J)+1
      GO TO 1
 14   CONTINUE
      GO TO 1
 17   XOBS(K)=XOBS(K)+
 1    CONTINUE
      PRINT 2 ,(XOBS(IK),IK=1,K)
 2    FORMAT(1X,*XOBS,S ARE =*,2515)
C-----COMPUTE XEXP(I), THE EXPECTED NUMBER OF DATA NALLING IN (I)TH G
C-----USE SIMPSON'S RULE TO INTEGRATE THE NORMAL DISTRIBUTION FUNCTIO
C     COMPUTE THE EXPECTED FREQUENCY OF DATA FALLING IN (I)TH GROUP
C-----ALAM IS THE STEP LENGTH USED FOR INTEGRATION USING SIMPSON'S RU
      XL=0.0
      K2=KM1/2
      DO 55 J=1,K2
      XA1=SMALL+FLOAT(J-1)*STEP
      XA2=(XA1-AVMU)/SIG
      XA=ABS(XA2)
      X=XL
      NN=IFIX((XA-XL)/ALAM)+1
      NK=NN-1
      DO 1 M=1,NN
      CALL FUN(YY,X)
      Y(M)=YY
 1    X=X+ALAM
      A=0.
      DO 2 I=2,NK,2
 2    A=A+Y(I)
      NL=NK-1
      B=0.
      DO 4 I=3,NL,2
 4    B=B+Y(I)
C-----AREA(J) IS TH  CUMULATIVE AREA UNDER NORMAL PROBABILITY DISTRIB
C     FUNTION TILL LAST ORDINATE OF (J)TH GROUP
      AREA(J)=(ALAM/3.0)*(Y(1)+4.0*A+2.0*B+Y(NN))+0.5
      IF(XA2.LT.0.0) AREA(J)=1.0-AREA(J)
 55   CONTINUE
      DO 57 J=1,K3
      KAB=2*J-1
      K3=K2+J
      K4=K1-KAB+J
 57   AREA(K3)=1. -AREA(K4)
C-----F(J) IS THE EXPECTED FREQUENCY OF DATA FALLING IN (J)TH GROUP
      F(1)=AREA( )
      DO 56 J=2,KM1
 56   F(J)=AREA(J)-AREA(J-1)
      F(K)=1. -AREA(KM1)
      DO 5 I= ,K
 5    XEXP(I)=FLOAT(N)*F(I)
```

```
      PRINT ..., (F(I), I=.,K)
  ..  FORMAT(.X,*F(I),S ARE  *,20F6.3)
      PRINT ..., (.XP(I), I=.,K)
  ..  FORMAT(.X,*X..P(I),S ARE *,15F7.2)
C     GROUP THE OBSERVATIONS INTO CLASSES WITH AT LEAST L OBS. EXPECT
C     CLASS
      K.=K+
      DO 6 I= ,K
      YOBS(I)= .
    6 YEXP(I)=0.
      J=.
      DO 8 I=.,K
      YOBS(J)=YOBS(J)+XOBS(I)
      YEXP(J)=YEXP(J)+XEXP(I)
      IF(YEXP(J).LT.FLOAT(L)) GO TO 8
      J=J+
    8 CONTINUE
C-----YOBS(J) IS THE NO. OF DATA IN THE (J)TH GROUP AFTER ABOVESAID R
C-----YEXP(J) IS THE NUMBER OF EXPECTED DATA CORRESPONDING TO YOBS(J)
C-----KNEW IS THE REAL DEGREE OF FREEDOM AFTER REGROUPING
      YOBS(J-.)=YOBS(J-.)+YOBS(J)
      YEXP(J-.)=YEXP(J-.)+YEXP(J)
      KNEW=J-.
      PRINT 103,KNEW,(YOBS(I),I=1,KNEW)
      PRINT 104,KNEW,(YEXP(I),I=1,KNEW)
  103 FORMAT(I5,20I5)
  104 FORMAT(I5,20F6.2)
C     COMPUTE THE CHI-SQUARE AND NO. OF DEGREE OF FREEDOM
      CHISQ=0.0
      DO 9 I=.,KNEW
      Z.=FLOAT(YOBS(I))
      Z.=YEXP(I)
    9 CHISQ=CHISQ+((ABS(Z.-Z2)-0.5)**2)/Z2
      NU=KNEW-.
      PRINT ..5,CHISQ,NU
  ..5 FORMAT(.X,*CHI-SQUARE=*,F.5.3,*      DEGREE OF FREEDOM=*,I5)
      PRINT ...
  ..6 FORMAT(... (.H-))
      GO TO 3
  ... CONTINUE
 9543 CONTINUE
      STOP
      END
      SUBROUTINE FUN(YY,X)
C-----THIS SUBROUTINE CALCULATES THE NORMAL PROBABILITY DISTRIBUTION
C     FUNCTION VALUE AT .
      Z=-X**2
      YY=EXP(Z/..0)/2.5065
      RETURN
      END
```

ERRATA

Page/Line	Printed as	Read as
iii/8	peoviding	providing
v/9	Production Line	Production Rate
8/11	Gufahr	Gutjahr
25/4	he BDR	the BDR
27/4	where, is	where, 1 is
40/11	V_i...	V_i as explained in section 4.2.2
		
46		
49	Call box 'DEPA	Call box 'DEPART'
50	Q_i	q_i
72/23	To confirm Chi,^{2}	To confirm this,^{2}

Date Slip

This book is to be returned on the date last stamped.

CD 6.72.9